Design • Build • Program • Learn • Compete

SERVO MAGAZINE

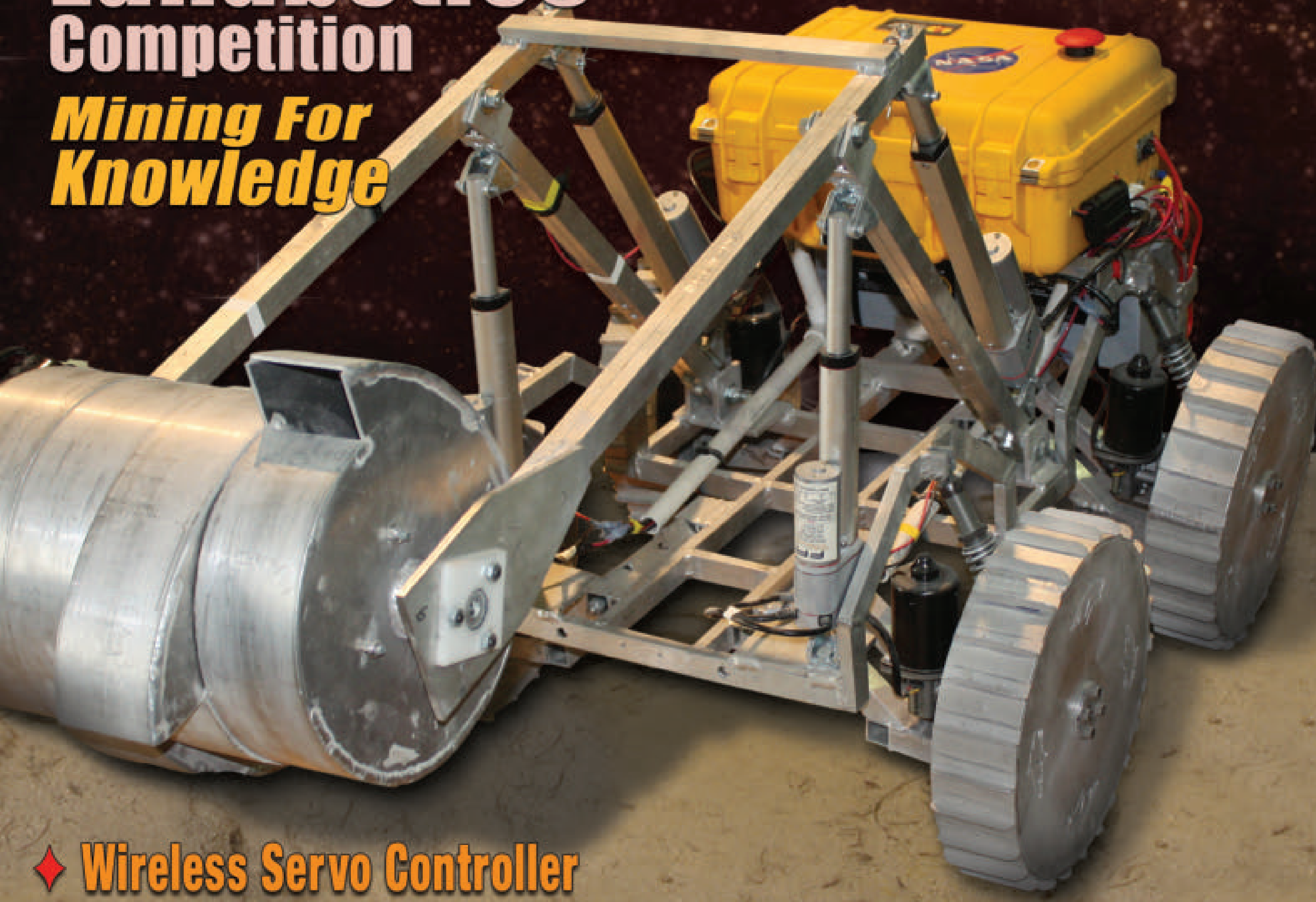
FOR THE ROBOT INNOVATOR

www.servomagazine.com

August 2012

3rd Annual Lunabotics Competition

Mining For Knowledge



- Wireless Servo Controller
- Touch Sensing
- Virtual Sensors
- Parrot AR Drone 2.0

SERVO editor Bryan Bergeron does a complete analysis, teardown, and mod of this popular quadcopter.

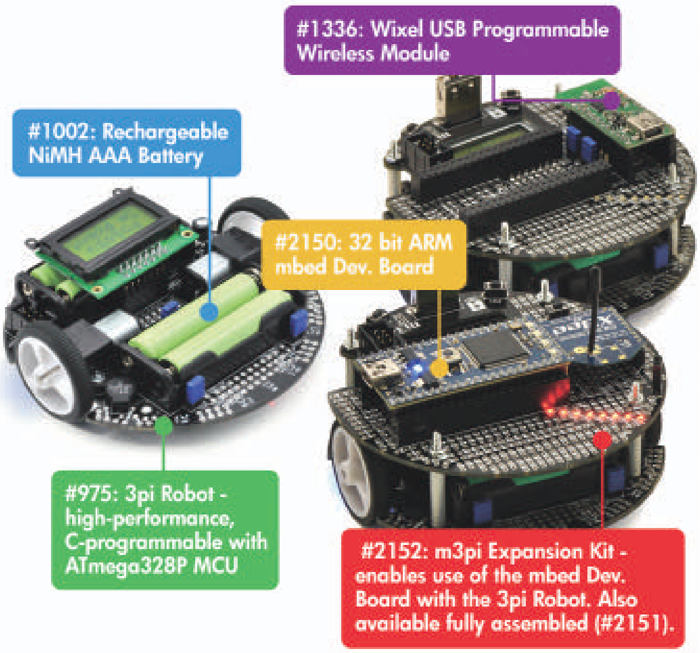
U.S. $5.50 CANADA $7.00

08

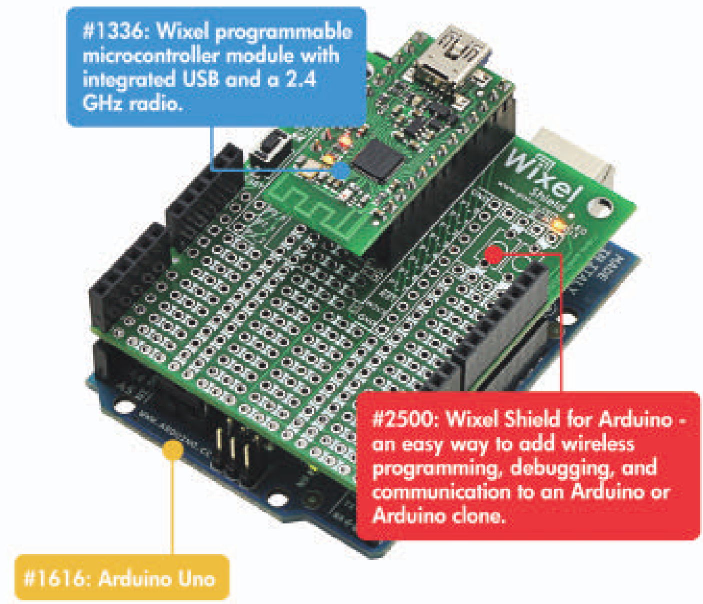
0 71486 02422 4

X1 AC PLUS X2 ULTIMA X2 400 Four Hundred X4 AC PLUS X4 80 Eighty

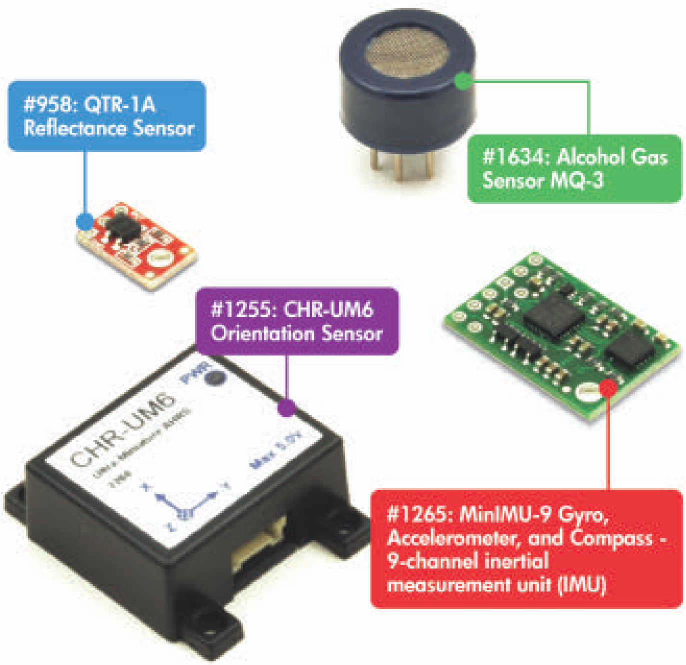
Robots and Robot Kits:
Pololu 3pi and m3pi



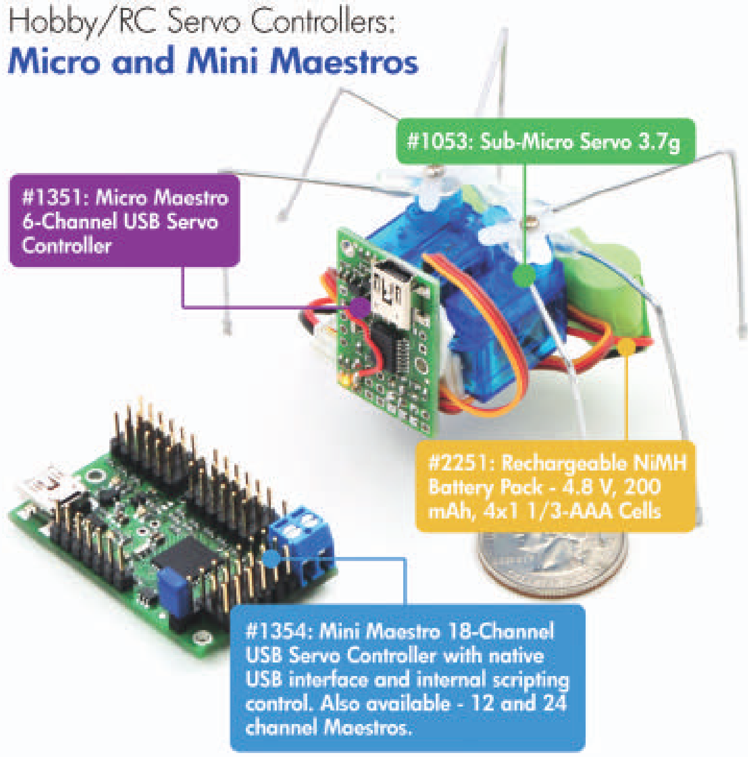
Programmable Controllers:
Wixel and Wixel Shield



Sensors:
Orientation, Reflectance and More



Hobby/RC Servo Controllers:
Micro and Mini Maestros



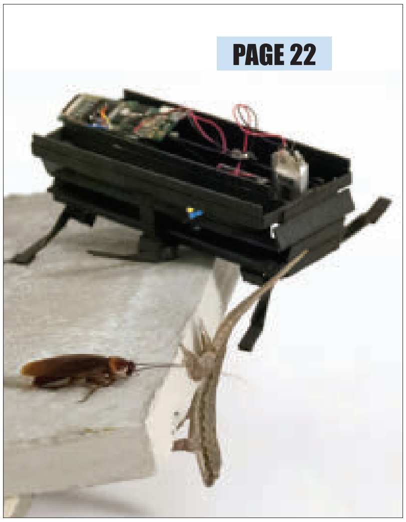
Finding the right parts for your robot can be difficult, but you also don't want to spend all your time reinventing the wheel (or motor controller). That's where we come in: Pololu has the unique products - from actuators to wireless modules - that can help you take your robot from idea to reality.

SERVO MAGAZINE

08.2012
VOL. 10 NO. 8

Departments

PAGE 22



The Combat Zone...

Features

PAGE 76

PAGE 10

Columns

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

In This Issue ...

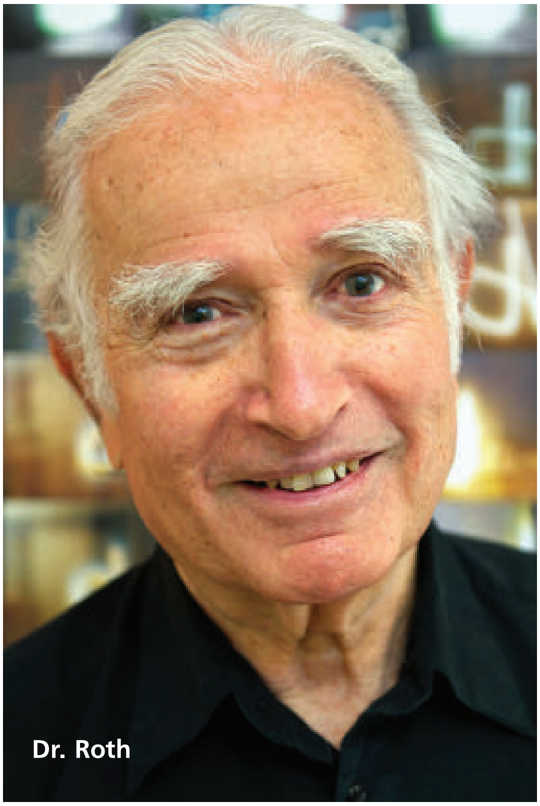
PAGE 72

PAGE 46

Mind / Iron

by Bryan Bergeron, Editor

2012 IEEE Robotics and Automation Award



Dr. Roth

The field of robotics wouldn't be where it is today without the work of bright, dedicated scientists and engineers. The best of the best contributors to robotics are recognized by the annual IEEE Robotics and Automation Award. The 2012 award went to Bernard Roth, Professor of Engineering at Stanford University, for "fundamental contributions to robot kinematics, manipulation, and design." Impressive jargon, but this doesn't say much about Professor Roth and his real part in making robotics what it is today.

According to Dr. Roth, he started out as an engineer interested in kinematics — the study of motion — and was talented at creating machines. One of the pioneers of the AI community at Stanford — the late John McCarthy — wanted a physical means of explaining his theories but had no experience building things. Roth and McCarthy began collaborating and the rest, as they say, is history.

I asked Dr. Roth to share his greatest achievement in robotics, expecting that he'd perhaps mention an algorithm or platform design. Instead, I was surprised to learn that when he first entered the field, it was full of one-offs — an arm here, a platform there, etc. — and that the creations were built without a theoretical basis. Dr. Roth managed to create the science of robotics — that is, to define the theoretical underpinnings of robotics that applied across all platforms and designs. He laid the foundation for manipulation, grasping, and other repeatable robotic operations we take for granted today.

When I asked Dr. Roth to speak to the future of robotics, he first went back in time, and reminded me that when robotics first emerged in the public arena the fact that they did anything at all was amazing, regardless of the robot size or complexity. Today, robots perform tasks elegantly, and we've come to expect this elegance. For the future, Dr. Roth envisions increased miniaturization, even more elegance, and more integration into everyday life in subtle ways. Eventually, he sees robotics disappearing into ubiquity.

As a career choice, Dr. Roth is bullish on robotics. He sees ever-expanding horizons without the dead ends common to other older and established fields. Now is an excellent time to enter the field because the horizons are unlimited.

I share Dr. Roth's enthusiasm for the future of robotics, as I'm naturally drawn to open-ended possibilities. Whether you're aiming for a degree in robotics or simply to get that carpet crawler kit working, we all owe Dr. Roth a round of applause for helping take robotics out of science fiction and into the world of engineering and hard science. **SV**

FOR THE ROBOT INNOVATOR SERVO MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Jenn Eckert, Tom Carroll, David Geer, Dennis Clark, R. Steven Rainwater, Kevin Berry, Gordon McComb, Pete Smith, Dave Graham, Andrea Suarez, Morgan Berry, John Blankenship, Samuel Mishal, Angel Hernandez, Bob Walker

CIRCULATION DEPARTMENT
subscribe@servomagazine.com

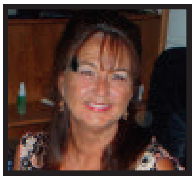
MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Shannon Christensen
Sean Lemieux


CHANEY ELECTRONICS, INC.
CELEBRATING 43 YEARS
We Offer Over 1,000 Electronic Components!
Our Full Inventory is Available on Our Website
Order Online with our Secure Shopping Cart System
http://www.chaneyelectronics.com/ ($10 Merchandise Minimum)
Orders Must Include Shipping
Toll Free: 1-800-227-7312 PO Box 4116, Scottsdale, AZ 85261
Visit Our Website for Full Details on Over 220 Electronic Project Kits and Science Labs
6 in 1 Solar Powered Boat Activity
$24.95 #C6983
Gain an Understanding of: Solar Power, Magnetism, and Motor Operation through: 6 Instructional Experiments and Building your own Solar Boat!
No Soldering Required! Great for Beginners!
Microcontrolled IR Remote Controlled Robot Kit
Build your own Robot and Control it with any IR Remote Controller!
Soldering Required
$59.95 #C6921
IR Vision Rover Robot Kit
Build your own Obstacle Detecting Self Navigating Roving Robot!
$35.95 #C6761
Soldering Required
26 in 1 Robotics Experimenter Lab
Gain an Understanding of Robotics Through: 6 Instructional Experiments and 20 Functional Robots!
Soldering Required
$52.95 #C6890
Lab Includes:
• All Electronic Components
• Detailed Color Manual
• Operational Theory, Schematics & Instructions for each Robot
100% Re-Usable
Light Spider Robot Kit
Build your own Light Tracking Self Navigating Robot!
Soldering Required
$12.25 #C6886
Chrome Dome Line Tracing Robot Kit
Build your own Path Navigating Line Detecting Robot!
$24.95 #C6900
Soldering Required
34 in 1 Robotics Exploration Lab
Gain an Understanding of Robotics Through: 6 Instructional Experiments and 28 Functional Robots!
Soldering Required
$71.00 #C6762
Lab Includes:
• All Electronic Components
• Detailed Color Manual
• Operational Theory, Schematics & Instructions for each Robot
100% Re-Usable
Little JitterBug Robot Kit
This is a Great Introductory Robot Kit for Beginners!
Soldering Required
Battery Included
$5.95 #C6984
The Unstoppable Bug Robot Kit
Build your own Wall Avoiding Self Navigating Robot!
Soldering Required
$18.95 #C6902

Robytes

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com.

by Jeff and Jenn Eckert

Navy Orders Minebots

The Knifefish mine-hunting robot, unveiled last April at the Sea-Air-Space Exposition.

It's not as if maritime mines are a new problem, as they have been around in crude form at least since the 16th century when a Chinese general (QI Jiguang) used them to fight Japanese pirates. Today, it is estimated that about 50 countries around the world stock more than a quarter million naval mines and could drop them into the oceans at any time. While a practical mine can cost as little as $1,000 to build, it is much more expensive (and dangerous) to retrieve and deactivate it. If you think minesweeping sounds like a perfect job for robots, you're right! The US Navy's answer is Knifefish: an underwater drone that will eventually be assigned to a fleet of littoral combat ships (i.e., small surface vessels that operate close to shore). On the outside, it looks much like a torpedo, but the Knifefish is a smart, autonomous, and sophisticated undersea minehunting bot. The 20 ft long robot uses low frequency broadband synthetic aperture sonar to locate mines and generate high-res images of detected objects, allowing it to tell the difference between real mines and most submerged debris. The bot operates for about 16 hours on a lithium battery charge, after which it returns collected information to the mothership. Eventually, the Navy hopes to deploy larger drones that can actually blow up encountered mines. Eight units are scheduled to be built by General Dynamics Corp. (**www.generaldynamics.com**) and Bluefin Robotics Corp. (**www.bluefinrobotics.com**), and are to be deployed by 2016. They're a little pricey at more than $20 million each, but that's a lot cheaper than losing a battleship.

The K-MAX unmanned cargo hauler lifts up to 6,000 lb.

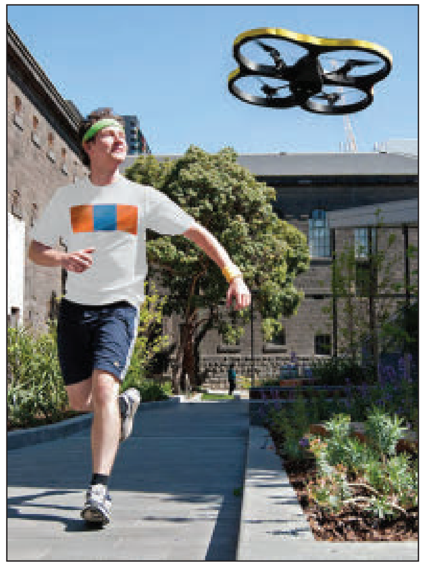
Drone 'Copter Sets Record

Developed in a joint project by Lockheed Martin (**www.lockheedmartin.com**) and Kaman Aerospace (**www.kaman.com/aerospace**), the K-MAX cargo hauler already had the distinction of being the first unmanned helicopter to deliver cargo and resupply troops in any combat zone; in this case, Afghanistan. Two units have been running six daily missions, delivering slings weighing up to 4,200 lb, with record deliveries totaling as much as 28,800 lb in a single day. After less than four months, the choppers passed the milestone of one million pounds delivered. According to Kaman, "The aircraft can carry more cargo on its four-hook carousel to more locations in one flight than any other unmanned rotary wing platform. Its intermeshing rotors eliminate the need for a tail rotor and allow for significantly improved lift performance and lower maintenance costs." The K-MAX can actually lift as much as 6,000 lb at sea level, but high altitudes in mountainous regions reduce the payload somewhat.

In addition, the company announced at the end of May that the K-MAX made aviation history by performing a "hot hookup," meaning that it can now swoop in, pick up military gear, and fly away without bothering to land. This capability could eventually eliminate the need for a human operator to control the vehicle.

Just FYI, Kaman Aerospace is a division of Kaman Corp., which also proffers products for missile and bomb safing and fuzing, high precision LIDAR and eddy-current sensing, specialty bearings and engineered products, and (what?) Ovation guitars.

Robotically Enhanced Exercise

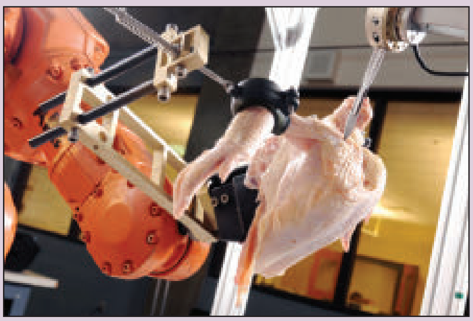


Joggobot aims to enhance the jogging experience.

A lot less brawny is the Joggobot, under development at Australia's Royal Melbourne Institute of Technology (**www.rmit.edu.au**). It isn't exactly an invention, though, as the platform is actually a foam-fendered AR Drone quadricopter which is really a flying game from Parrot SA (**www.parrot.com**) that can be controlled using an iPhone, iPod Touch, iPad, or (soon) other mobile platforms. You can pick one up for about $300 from Amazon and other vendors. The difference is that Joggobot incorporates custom software that enables you to enter your preferred jogging speed into a smartphone app. The app controls the drone so that it flies ahead of you at the proper pace. Alternatively, you can program it to maintain a set distance regardless of how fast you are running. It accomplishes that by monitoring a blue-and-orange pattern on the t-shirt that you are required to wear.

Noted the developers, "We ask questions such as `Should the robot be a pacemaker for the jogger? If so, can this be motivating? Or should the Joggobot be more like a dog, reacting to the jogger like a companion? How does this affect the interaction and, in particular, the exercise experience for the jogger?'" Another question one might ask is, "While you're focused on the Joggobot, what's to keep you from jogging in front of a moving bus or stepping into an open manhole?" Guess it could provide endless amusement for those of us who are just watching.

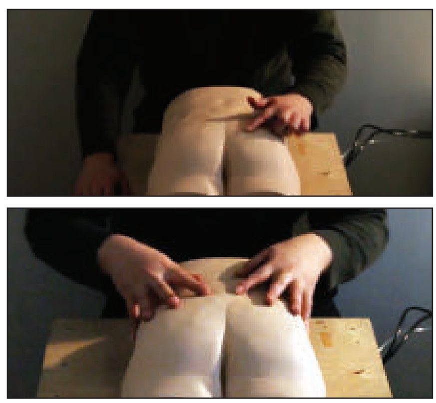
Fowlbot Automates Deboning



Robotic system prepares to slice through the shoulder joint of a chicken.
Photo courtesy of Gary Meek.

One doesn't usually think of the Georgia Tech Research Institute (GTRI, **www.gtri.gatech.edu**) as being in the chicken business, but it turns out that poultry is Georgia's no. 1 agricultural product with an estimated annual economic impact of nearly $20 billion. Accordingly, researchers have created the Intelligent Cutting and Deboning System which uses advanced imaging technology and a robotic cutting arm to automatically debone chicken and other poultry products. The system employs a 3D vision system that determines the proper cutting procedure, allowing the device to automatically butcher the bird so as to optimize yield and reduce the risk of leaving bone fragments in the product. In case you're interested in the details, this prototype system uses a fixed two-degree-of-freedom mechanism for making simple planar cuts. The bird is mounted on a six-degree-of-freedom arm that allows alignment of the bird to any desired position. The robot arm places the bird under the vision system and moves it with respect to the cutting robot. The system employs a force-feedback algorithm that can detect the transition from meat to bone which allows the knife to slice through ligaments while avoiding the bone. In addition to minimizing bone chips, the system promises to increase product yield. As noted, "Every one percent loss of breast meat represents about $2.5 million to each of Georgia's poultry processing plants." Somewhere Frank Perdue is smiling.

Breakthrough in Robutties



Whip it or caress it, it's Shiri the robobutt.

If it's been bad, you can spank it. If it's been good, you can caress it. If you're just weird and perverted, you can poke it with your finger. In any case, it will respond by expressing "various emotions with organic movement of the artificial muscles." "It" is Shiri — a pair of humanoid haunches made of a rigid urethane skeleton encased in silicone skin. Each cheek contains an airbag that employs modulating pressure to generate various reactions to your advances, including "quivering in fear." And, yes, you can see it in action at **www.youtube.com/watch?v=hkIJGjXoBAA**. Taking credit for this feat of engineering is Nobuhiro Takahashi of Japan's University of Electro-Communications (**www.uec.ac.jp/eng**). His parents must be so proud. **SV**

GEERHEAD

by David Geer

Contact the author at geercom@windstream.net

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Students Get a Bot Powered Kick Out of Mathematics

Something fun, robotic, and mathematical is happening in Gulf Stream, FL, located on the state's southeastern coast. That's where Shawn Harahush — sometimes referred to as RoboTeacher — uses robots, engineering, and rockets to teach math to students at the Gulf Stream School. By turning math into hands-on projects, Mr. Harahush challenges his students, who respond remarkably.

The Learning Process, Technically Speaking

In the school's Robotics program, Mr. Harahush uses teaching methods that combine science, technology, engineering, and math, quadrupling class coverage areas and turning the subjects into practical lessons that make learning tough subjects a thrill. "These methods infuse instruction with key concepts such as logic, trial and error, and proportional reasoning," explained Mr. Harahush. The coursework is age-appropriate, despite the seeming complexity and depth, and it appeals to girls and boys alike.

In this STEM (Science, Technology, Engineering, and Mathematics) program that the school created (more on that later), students create solar cars, T-Bots, and LEGO Mindstorm NXT kit robots that promote skills in estimating angles, distances, and elements of time while inputting critical data for the robot's operation. "Testing their predictions enabled the students to become more accurate with their measuring skills," Mr. Harahush continued.

Because the students are familiar with LEGOs, they find using LEGO Mindstorm kits to build robots very easy and natural. While the students construct the physical robots quickly, it is the robot programming that gives their thought processes an exercise. "Programming the robots to

Gulf Stream School students and their teachers: (front) Timothy Lynch, Helen Huisinga, Kiara Warren and Isabel Pearce; (back) Joey Morfogen, Ingrid Marinak, teacher Glenn Harland, Connor Hopkins, Christopher Klein, Jack Young, teacher Shawn Harahush, and Maddy Uible.

RoboTeacher Shawn Harahush assists students Ingrid Marinak and Joey Morfogen while surrounded by other students and teacher Glenn Harland.

complete basic tasks such as following a black line around a track or driving a certain distance and turning challenges the students," says Mr. Harahush.

The LEGO software helped to simplify programming concepts for the pupils by using LEGO blocks that natively serve a given function or purpose to meet the class assignment's objective. (Of course, the instructors help the students make the connections between one concept and the next, so they can leap forward in their learning.)

It is the problem solving that fully involves the students as they work together in teams to build their robots. Students in the seventh grade, for example, would meet for 40 minutes once a week over a 12 week period during study hall. There were no requirements for those periods except to make the robots work. "If the robot did not work as the students hoped, they examined the problem, found the solution, and were excited to see their ideas come to life," Mr. Harahush detailed.

Student Project Specifics

Building T-Bots using the Pitsco Education T-Bot II Hydraulic Arm project, the students dove "arms" first into the worlds of hydraulics and mechanics. These endeavors taught them about determining range of motion, axis, programming, and team building.

Using the color sensors that the LEGO robotics kit included, the students built robots that could distinguish colored objects and follow colored (black) lines on a track. The students learned to think with conditional reasoning, performing objectives in a repeated manner using loops (examples include sorting different colored balls or Skittles).

To hone their skills in trigonometry, the students built Estes rockets that they could launch which taught them complex math and how to calculate the highest altitude to which the rockets could soar. "We used protractor-like instruments to measure angles. And since we knew the distance between the launch point and the launch pad, we could use basic trigonometric rations to approximate the rocket's peak height," Mr. Harahush commented.

While not robotic, rocket skills could come in handy some day should the students take a queue from Iron Man and build their own rocket flight-enabled exoskeletons.

Though time did not permit the teachers to lead the students in the bridge building portion of this curriculum, they plan to teach it and cover an introduction to construction, forces, and structures, giving students a physical view into the challenges that architects see every day. "They will learn that bridge design and building is much more than a simple matter of aesthetics," Mr. Harahush asserted.

Isabel Pearce and Kiara Warren discuss their robotic project with teacher Shawn Harahush.

Fast Facts

Under Mr. Shawn Harahush's tutelage, Gulf Stream School students in the fifth through eighth grades achieve a lot, including:

- Building bridges and hydraulic arms.
- Constructing robots enabled with touch and color sensors.
- Developing a security bot.
- Learning trigonometry through rocket building.
- Acquiring engineering skills through robot building.
- Participating in the STEM program.

Resources

Gulf Stream School, South Florida
www.gulfstreamschool.org

Results

The efforts of the school, teachers, students, and parents in surrounding the young roboticists with both book learning and building skills are paying off in great grades and advanced preparation for higher learning. Mr. Harahush's own enthusiasm mirrors that of his students as he discusses their accomplishments:

"Because each group was at a different phase of construction and programming, every day brought excitement to the classroom. When one group advanced and the others saw the robot completing its tasks, it gave them new ideas and spurred them on creatively. From the beginning, students were enthused having seen some final products in action on the YouTube and LEGO sites. As every piece and section clicked into place, anticipation grew as they approached their goals."

By combining the four disciplines in STEM, the school engaged the students in a multi-disciplinary array of topics that gave them a glimpse into what a profession in robotics or engineering would involve. By concentrating on confidence building rather than grades (no grades are given in this particular program), the faculty helped the students acquire real world tools for overcoming potential future challenges — whether academic or technical.

The program gave the students increased estimation skills, a strong grasp of concepts, and the beginnings of a math-based prowess in perceptions of time, distances, and angles.

STEM — the Root of the Solution

This STEM program was a joint venture between the math and science departments at the Gulf Stream School. The idea was to integrate what they each taught in their own prospective disciplines into something that was more than the sum of its parts, and more than what the existing curriculum could handle.

By developing a cross-curricular course offering students a chance to undertake activities that are challenging, relevant, and pertinent to the present world, the faculty gave the students the opportunity to problem-solve both independently and as part of a team. These technical activities made the students better team members and problem solvers, endowing them with skills they will use in their future academic and professional lives.

"The program made them more aware of the connections between math, science, and the world around them. Hopefully, it inspired them to pursue a technical, engineering-based vocation," Mr. Harahush concluded.

Final Thoughts

A recent group of Gulf Stream students showcased their robots for an audience of teachers, parents, and student peers. With the program and course work a complete success, the school hopes to expand on it for future sessions. SV

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Our resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

by
Dennis Clark

ASK MR. ROBOTO

As you read this, it is August, the summer is ending, and it is time to get back to the design boards for next year's summer robot season. You will have been to some big expos and competitions, and will have long lists of ambitious activities to make your robot even better, stronger, and faster. I want to get my RoboBuilder robot back up and running with his new hip servos installed so that he can moon-walk when he dances (okay, so that he can WALK period!). Some of my non-Windows buddies and I are looking at a different controller backpack with a bit more speed, power, and capability than the stock one. The Rasberry PI is on that list, but in my case, a Digilent Cerebot controller seems to be a good option. For Robo One competitions, dynamic stabilization is the golden solution, but just motion will make me happy for now! At the other end of the expense continuum, I am building a Mech Warrior style robot from Lynxmotion system parts that will fire LEGO darts at intruders (like my cat), which will be fun. How many of you out there are trying your hand at walkers? Rolling robots are great for many things, but nothing beats a walking robot for crowd appeal!

This month, I'm going to wrap up my series on the Digilent MAX32 board and MPLAB development. I'll add a serial port communication back to the PC so we can see what is happening, while showing a fast way to send numbers — not just static text — over an ASCII communications link. We'll see what else we can do before I run out of time!

The Last Installment on Programming the Digilent MAX32

Last month, we ended by implementing an *Input Capture* module to measure the pulse given by the HC-SR04 sonar module. There was no really good way to blink an LED to approximate the distance measurement, so I simply left a spot where you could stop the debugger to get a value. That was a bit clumsy, so this month we'll implement a virtual COM port using the USB connection to the MAX32 board. When I set up the Peripheral Bus Clock on the PIC32 way back when, I set it to 10 MHz to make my math dealing with timers easy. This is the clock that will be used with the USART on the PIC32. Just to pick a number, I chose a bit rate of 38400. Also — just to pick a formula — I chose to use the BRGH (High speed Baud Rate Generator) = 1. In Section 21: UART, document number DS61107F, Section 21.3, we see that the baud rate calculator formula for BRGH = 1 is:

$$UxBRG = \frac{F_{PB}}{4 \bullet BaudRate} - 1$$

where F_{PB} is the peripheral bus clock (10 MHz) and UxBRG is U1BRG (the baud rate generator register for UART1). Doing the math, we see that U1BRG would be 64.104. We can't use fractions in the BRG

Listing 1: Set up the UART1 to talk to a COM port.

```
void SerialInit(int baud)
{
    /*
    * 38400 baud is good for a 10MHz PBUS.
    *       using BRGH = 1, baud, 8N1
    *       Directly set bits, too many to use "OR'd" bit fields!
    */
    int n;

    U1MODE          = 0x00008008;
    U1STA   = 0x00005400;
    U1BRG   = (PBUSFREQ/(4*baud)) -1;

    // Clear the screen
    for (n=0;n<80;n++) {
            putsUART1("\r");
    }
}
```

register, so we round down to 64. We want our baud rate to be as close as possible to what it should be so that there won't be any bit error rate. I've found that anything over about 1.5% can be an issue, so let's check what we'll get with a BRG of 64. We'll use this formula (also found in section 21.3) to check our final baud rate:

$$BaudRate = \frac{F_{PB}}{4 \bullet (UxBRG + 1)}$$

My math says that the final baud rate ends up at 38462 which is about 0.16% off. No problem. **Listing 1** shows how we set the UART up in our project.

In this case, since there are *so* many options that can be chosen to set up a UART, I chose to set bits directly to the configuration registers instead of using the 32-bit peripheral library; it just got too tedious to OR a bunch of bit values. **Figures 1**, **2**, and **3** show the layout of the three registers I set up to get what I wanted.

The two bits that I set in the U1MODE register are ON and BRGH. These turn the UART on, and tell it to use the high speed BR generator.

The three bits I set in the U1STA register are '01' in the UTXISL which says to issue an interrupt when all characters have been transmitted. I was unsure if I was going to use interrupts, so I set this up. It turns out I didn't use transmit interrupts. I also set the UXREN bit which enables the UART receive block. Finally, I set the UTXEN bit which enables the UART transmitter.

I let the compiler pre-processor handle the math for setting the U1BRG register. Done this way, if I change the PB clock all I need to do is change the #define in the program and the BRG is set correctly without me needing to fiddle with it. Hopefully, it comes up with a BRG divider that has a low error rate.

Okay, now to configure the USB virtual COM port on the MAX32

Register 21-1: UxMODE: UARTx Mode Register[1,2,3]

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 31							bit 24
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ON	FRZ	SIDL	IREN	RTSMD[4]	—	UEN<1:0>[4]	
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL<1:0>		STSEL
bit 7							bit 0

Legend:
R = Readable bit W = Writable bit P = Programmable bit
U = Unimplemented bit, read as '0' -n = Bit Value at POR: ('0', '1', x = Unknown)

Figure 1. UXMODE register.

Register 21-2: UxSTA: UARTx Status and Control Register[1,2,3]

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	ADM_EN
bit 31							bit 24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADDR<7:0>							
bit 23							bit 16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-1
UTXISEL<1:0>[4]		UTXINV	URXEN	UTXBRK	UTXEN	UTXBF	TRMT
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/W-0	R-0
URXISEL<1:0>[4]		ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

Figure 2. UXSTA register.

Register 21-5: UxBRG: UARTx Baud Rate Register[1,2,3]

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 31							bit 24
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG<15:8>							
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG<7:0>							
bit 7							bit 0

Figure 3. UXBRG register.

Listing 2: Main sonar/print output loop.

```
Init();
SerialInit(38400);
putsUART1("Ready\r");

for (n=0;n<4;n++) t[n]=0;
/*
* We have set everything to interrupt when two edges have been
* seen and recorded.  The mIC1CaptureReady macro only tells us
* that at least one edge has been recorded.  The IC hardware buffer
* is four deep, we don't care about the first (rising) edge of the
* echo pin, only when it falls, so we need to look at the second
* edge recorded.  Then we'll do our calculations to get our ranges.
* Calculations:
* Subtract 200us from the second edge count (200us/800ns = 250 tics)
* speed of sound is about 34000cm/s, each TMR3 tic is 800ns so
* each tic is (34000 * .8us) about 1/37 cm out and back, so
* tics/37 is cm out and back.  Divide this by 2 and you have cm to
* the target.  To get inches, 37 * 2.54 is about 93.  You can do
* more precise math to get better resolution that this, but this
* is good enough for an object detector.
*/
while (1) {
        mIC1ClearIntFlag();
        mPORTFSetBits(BIT_1);
        for (delay=0; delay<15; delay++);
        mPORTFClearBits(BIT_1);

        TMR3 = 0;

        // Wait for Capture events
        while (!IFS0bits.IC1IF);
        //Now Read the captured timer values
        n=0;
        while(mIC1CaptureReady()) {
                t[n++] = mIC1ReadCapture();
        }
        cm = (t[1]-250)/37/2;
        in = (t[1]-250)/93/2;
        putsUART1("SONAR Range\r");
        if (cm > 250) {
                cm = 999;
                in = 999;
        }
        putInt(cm);
        putsUART1(" cm\r");
        putInt(in);
        putsUART1(" in\r\r");

        delay = t_1ms + 1000;
        while(t_1ms < delay);
}
```

board, we just call the *SerialInit()* function with the desired baud rate and we're done. On your computer, you'll have to configure a terminal program to the correct baud rate to get the data. I use a Mac, and on my machine I used Z-Term and pointed it to /dev/tty.usbserial-A6009D1L which I admit, isn't all that obvious. However, Z-Term found it (without the tty. prefix). Those of you that use Windows machines will probably have this virtual COM port show up as COM5 or something similar. You'll note that the last step in *SerialInit()* is to issue 80 `putsUART1("\r");` statements. I did that to clear the screen of the terminal. I didn't know what command character would be needed to do that, so I did it the "brute force" way of forcing everything off the screen.

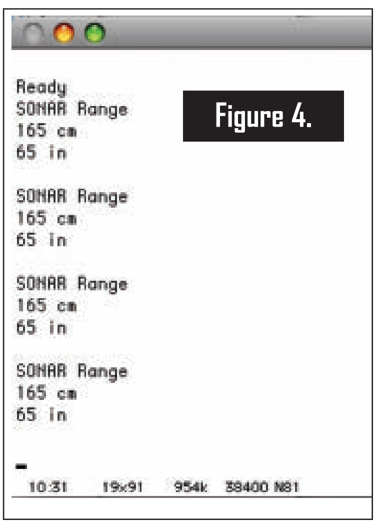


Figure 4.

Now that we have the UART set up and we've configured our terminal emulator to listen to the USB/serial port, we need to send stuff. Sending a string is pretty simple; we'll use the 32-bit peripheral library UART calls to do the heavy lifting. In **Listing 2**, I show how we do that. This is the main loop that does the sonar work and prints out the data.

In this code, we start by initializing the PIC32, then the UART to 38400; then, we clear the array holding the timing locations of every pulse edge that the IC1 module detects, starting with the rising edge of the *echo* line. We're only interested in the second edge (falling) that tells us when an echo was seen by the HC-SR04.

The loop that captures these uses the peripheral library call that checks the module for values captured. We can only read these in order by calling the Read Capture every time the hardware module tells us there is another one ready to read. We know to start looking when the IFS0bits.IC1EF bit tells us that we have two edges recorded. (See the source code; we told the IC1 module to interrupt when it had captured two edges, starting with a rising edge.) Now that we have the measurement we want, we convert it to centimeters and inches, and send it out the serial port. Interestingly, the peripheral library does not have a conversion routine for binary to integer — or any other number type — so we'll have to do that ourselves. This is really easy, and **Listing 3** shows how this can be done.

This routine will convert any 32-bit number to a printable value and send it out the serial port. I don't want any leading zeros, so I start with the variable *msb = 1*; this means that no digit has been printed yet, so don't print a zero if the result of the place divide comes up a zero. As soon as any non-zero digit is printed, then it is okay to print zeros (*msb = 0*). In the end, we need to check for the special case where the number is indeed zero.

Listing 2 shows how we figure inches and centimeters, so I won't do it again. When we run our program, every second the MAX32 board will print the sonar range to an object in centimeters and inches to the serial port as in **Figure 4**.

I don't know about you, but I always get a little thrill when I see a project come together! **Figure 5** shows what my lash-up looks like. Although it might be sacrilege to say so, I don't really like the Arduino I/O pinout. It doesn't put power and ground on the same connector as the I/O lines which makes it difficult to build a connector for a device. I did find something called a *Sensor Shield* for the Arduino MEGA at **http://arduino-direct.com** which has exactly what I want (and just went onto my wish list!). It doesn't support all of the I/O that the MAX32 has, but it is a good start to be able to use simple wiring connectors for your sensors on some of the MAX32 I/O lines.

Listing 3: Converting and printing an integer value.

```
void putInt(unsigned long val)
/*
*       Print out an unsigned integer to UART, no leading zeros.
*       val = integer to send
*/
{
        unsigned long temp  = val;
        unsigned long place = 1000000000;  // start at 1 billion
        unsigned char c;
        unsigned char msd      = 1;        // most significant digit

        while (place > 0) {
                c = (unsigned char) (temp/place);
                if (c > 0) {
                        msd = 0;                        // no longer on MSD
                      // remove this from the total
                        temp-= ((unsigned long)c)*place;
        }
                if (!msd) {
                        putUART1('0'+c);
                }
                if ( (place == 1) && (temp == 0) && (msd == 1)) {
                        putUART1('0');                  // number was zero
                }
                place/=10;                              // To next 10's position
        }
}
```

If anyone has any other questions about this useful little board, please send me an email and I'll be happy to look into it. I know that the chipKIT boards will be on my experimenter's list for a while; I've yet to do more than touch on their capabilities. In the meantime, at the article link you'll find a zip file called robotoAugust.zip.

Well, I've used up another chunk of time having fun with your questions, and wrapped up my (unintentional) programming series on the chipKIT MAX32 board and MPLAB programming for it. I have not yet gotten a chance to build a project for use with the Arduino bootloader for this board, but eventually I will.

Remember, you can contact me at roboto@servomagazine.com to ask any questions about robotics that you have. I'm here to answer them!! **SV**

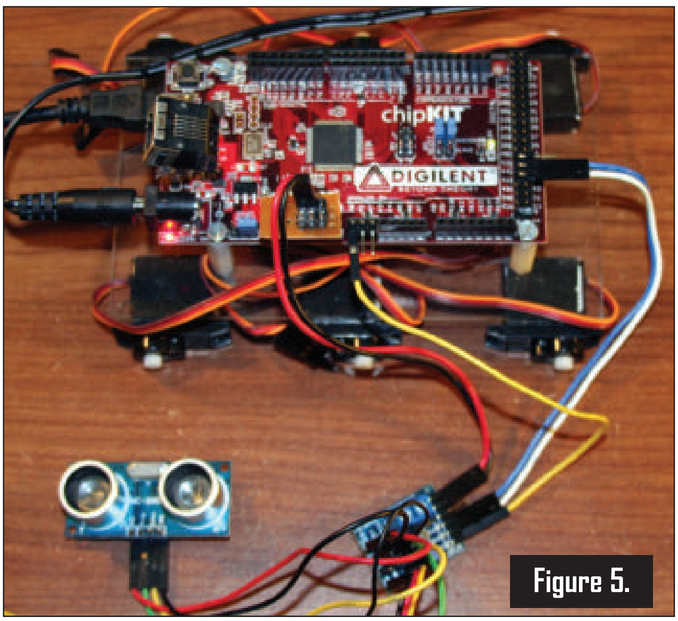
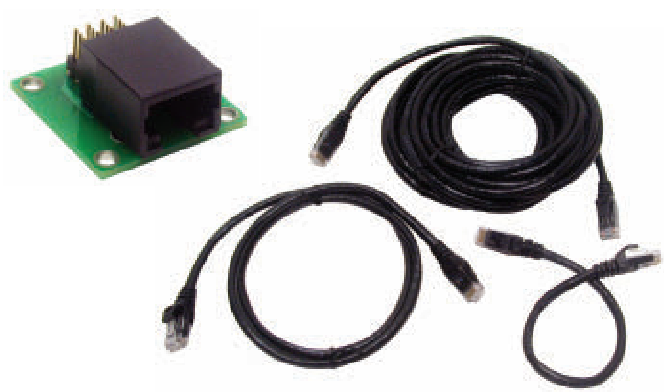


Figure 5.

NEW PRODUCTS

New CAT6 Cable and Adapter

ServoCity now offers a clean and simple way to extend servo wires using a CAT6 cable. The new CAT6 adapters with signal booster allow the user to plug up to four servos into the row of pins. The servo wires will be extended through a single CAT6 cable and then converted back into



servo pinouts in order to plug into any standard servo controller. The gold plated CAT6 connectors and high strand-count wires ensure that there is very little resistance between the controller and the servos. The cables and adapters can replace multiple standard servo extensions to provide a clean and efficient way to wire up a robot, pan and tilt, or any other project that requires extending the wires between the controller and the servo.

Open Loop Pan and Tilt System

ServoCity is also introducing its new MPT1100-SS pan and tilt system. The pan and tilt head is the ideal size for DSLR and smaller bodied cameras. The super light 6061 T6 aluminum structure weighs in at 2.2 lbs, and is able to handle up to a six pound payload. Dual ball bearings on each axis support the 1" hollow aluminum shafting, allowing you to route camera wires through the pan and tilt head. The head easily attaches to a jib crane in the upright or hanging position without any tools.

The MPT1100-SS is precisely controlled by a new digital joystick controller. The two-channel controller is fully proportional and the maximum speed of each axis can be limited by the min/max knobs for ultra-slow shots. Setup is quick and easy as the controller links to the pan and tilt head with a single CAT6 cable and is powered by a 12V power supply. The joystick includes a jib crane mount so you can easily attach it to any 1" OD bar. The system ships as a ready-to-run unit that includes a joystick control, 25 foot CAT6 cable, and 12V power supply. The fully assembled system is $649.99.

For further information and detailed schematics, please contact:

ServoCity	Website: **www.servocity.com**

Firefly Device

Solarbotics is introducing its new Firefly which has a coin-cell powered microcontroller that is designed to run three LEDs through button-selectable light blinking sequences. It has smooth transitions and five different modes (Blink All, Metronome, Disco, Fading Eyes, and Firefly) with three speeds for each mode. In Firefly mode at regular speed, a single battery will last more than 24 hrs.

Not much larger than a coincell battery, this tiny microcontroller-based device is a wearable and hackable through-hole kit, and a perfect introduction to soldering. The manual can be downloaded from Solarbotics' website, and it guides users through the simple steps for assembly. Firefly's totally controllable lighting sequences can add some flare to name tags, make a blinking lantern for special holidays, or provide a luminous night-time event by putting them into jars to float in water.

If your robot needs some more personality, add the Firefly to give it glowing eyes. Or, if used with a magnet, it can be a smart LED throwie. It's also wearable; an extra eyelet is included on the PCB to tie onto a necklace, key-chain, ring, or bracelet. You can also sew it onto clothes with some conductive thread.

The Firefly not only blinks and controls LEDs, but it's also a fully capable development board for the ATtiny 24/45/85 series of microcontrollers. A huge feature of the Solarbotics FireFly is that it has the capability of being re-programmed in the Arduino 1.0 environment. Using a small modification published by MIT Media Lab's High-Low tech blog, there are now several tutorials across the web showing how to do this for yourself.

The board has a cleverly hidden three-pin Ground-Voltage-Signal (GVS) expansion port that makes it usable for project expansion. The code for the Firefly is completely open source and has some great examples on using Arduino's random, interrupt, and sleep functions, as well as faking hardware PWM with "For" loops. The device has an operating voltage of 2.0-5.5 VDC, runs off of its 8 MHz internal oscillator, and in sleep mode it will draw only 200 nA (that's 0.0002 mA). Potentially, the Firefly's battery can last for more than a year in sleep mode.

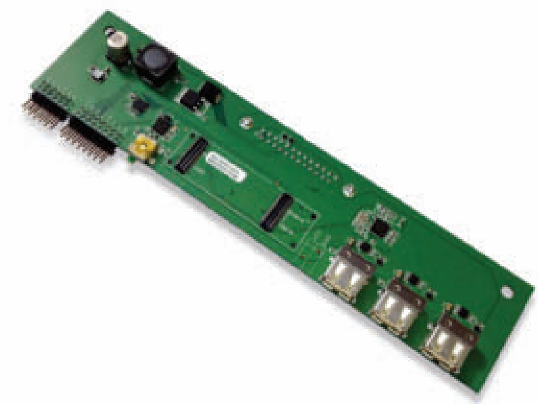
For further information, please contact:

Solarbotics, Ltd. Website: **www.solarbotics.com**

TurtleCore Expansion Board

Gumstix, Inc., is now offering the TurtleCore™ — a new, low cost expansion board that will enable robotics enthusiasts to build robots easily and cost-effectively using the iRobot Create platform. The TurtleCore expansion board — in combination with a Gumstix Overo® or Overo STORM series COM — integrates directly into the iRobot Create, eliminating the need for a netbook to operate the mobile robot or interface with camera inputs.

The TurtleCore expansion board connects directly to the Create via DB25 port, drawing power for and providing control to an Overo COM. This fully functional computer



Low Cost SCARAs Line Expanded

EPSON Robots has expanded its low cost LS-Series SCARAs. This new robot series is available in arm lengths of 400 mm (LS3) and now 600 mm (LS6).

LS-Series SCARA robots were created at a reduced cost solution for factories looking for maximum value without giving up performance.

Some of the unique characteristics of EPSON LS-Series SCARA robots include:

- Fast Cycle Throughput
- Industry Leading Ease of Use
- Low Cost and High Performance
- Compact Footprint Robots
- ISO 4 Clean Compliant Models Available

Manufacturers who would normally explore the option of low cost Cartesian type linear axes robots for automation are now able to consider the use of LS-Series SCARA robots as an affordable and high performance alternative.

For further information, please contact:

Epson Robots Website: **www.epsonrobots.com**

includes three USB 2.0 ports and has a form factor tailored specifically for the Create (where alternative computing platforms may not fit).

The TurtleCore expansion board offers several advantages to Create developers including:

- Fits in the cargo bay of the Create programmable robot with room to spare for sensors, a bigger battery, or even additional COMs.
- Draws power for the COM directly from the Create, so there is only one battery to charge. The inherent low power requirement of Overo and Overo STORM COMs coupled with software control of the Create power system ensures maximum range.
- Three USB ports and two breakout headers for useful signals from the Create and the Gumstix COM simplify the addition of sensors and actuators. Also includes standoffs and screws for secure mounting.

The TurtleCore expansion board retails for $89. For further information, please contact:

Gumstix, Inc.	Website: **www.gumstix.com**

EVENTS Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**.

— **R. Steven Rainwater**

AUGUST

4 **Chibots SRS Robo-Magellan**
Moraine Valley Community College, Palos Hills, IL
Autonomous robots compete on an outdoor course that requires waypoint navigation and vision.
www.chibots.org

9-19 **Missouri State Fair Robot Expo**
Sedalia, MO
Various robot events including the 4-H Show-Me Robotics Competition. Check the Fair schedule for exact days and times of each event.
www.mostatefair.com/special-events

20-25 **FIRA Robot World Cup**
Bristol, England
All the usual autonomous robot soccer divisions, ranging from tiny robots up to humanoid bots.
www.fira.net

31 **DragonCon Robot Battles**
Atlanta, GA
Remote controlled vehicles destroy each other at the DragonCon science fiction convention.
www.dragoncon.org

SEPTEMBER

3-8 **National Junior Robotics Competition**
Science Centre, Singapore
This robotics competition encourages students to develop problem solving skills, entrepreneurial skills, creative thinking skills, and team spirit among the participants.
www.science.edu.sg/events/pages/njrcompetition.aspx

15 **Robotour**
Czech Republic
Autonomous navigation in a park carrying a five liter barrel of beer.
www.robotika.cz

17-21 **World Robotic Sailing Championship**
Cardiff, Wales, UK
Robot sailboats must navigate an ocean course around buoys.
www.roboticsailing.org

21-23 **RoboCup Junior Australia**
Canberra, Australia
RoboCup Junior Australia is a project-oriented educational initiative that supports local, regional, and international robotic events for young students. Teams work in a co-operative and supportive environment in three distinct challenges: Dance, Rescue, and Soccer.
www.robocupjunior.org.au

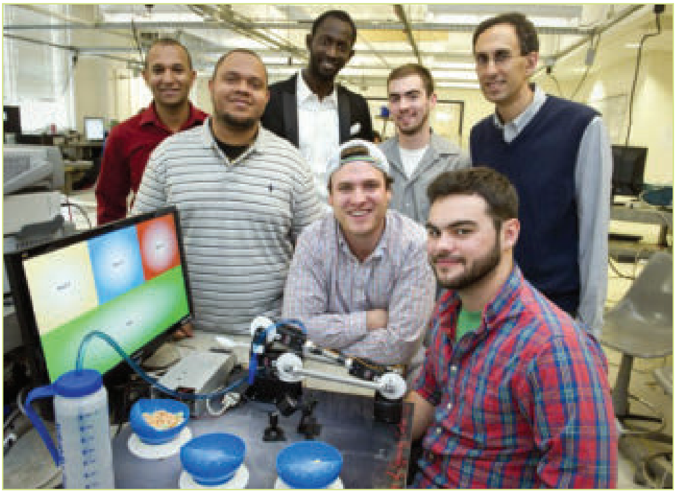
OCTOBER

1-4 **UAV Outback Challenge**
Kingaroy, Australia
Search and Rescue Challenge, Airborne Delivery Challenge, and Autonomous.
www.uavoutbackchallenge.com.au

18-21 **Latin American Robotics Competition**
Fortaleza, Brazil
Events include the Brazilian Robotics Competition, Robocup Latin American Open, and Brazilian Robotics Fair.
www.cbrobotica.org

19-21 **Critter Crunch**
Hyatt Regency Tech Center, Denver, CO
Robot combat — 2 lb and 20 lb event catagories. Autonomous and remote control. Starting size of 12" x 12" x 12". Weight limit of 20 lb. Power source must meet OSHA requirements for indoor use.
www.milehicon.org/?page_id=16

bots IN BRIEF



iCRAFT (eye Controlled Robotic Arm Feeding Technology) won first place in ECE Capstone Design Competition 2012. Pictured are teammates Ryan LaVoie, James Barron, Pedro Lopes, Nick Aquino, Basel Magfory, Mohammed Kante and advisor electrical and computer engineering professor Waleed Meleis. Photo by Mary Knox Merrill.

THE EYES HAVE IT

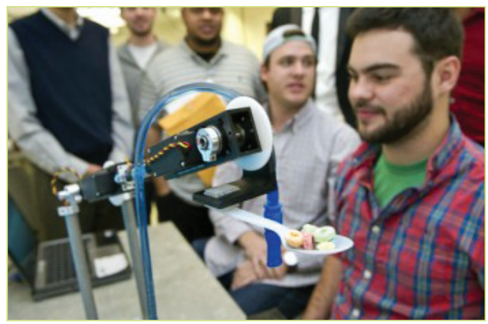
A Northeastern University team has recently developed an Eye-Controlled Robotic Arm Feeding Technology that can be manipulated by vision. The iCraft was designed for the disabled or elderly to aid them while eating. By looking toward the desired food, the robotic arm will take over. The prototype was built for approx. $900, and the team has released open source software for those who would like to build their own.

There's no right pace," said Mohamed Kante, who worked with elderly and disabled patients at Kindred Transitional Care and Rehabilitation in Fall River, MA. No matter how fast or slow he and his colleagues offered patients bites of food, they could never match the patient's individual needs.

So, Kante and five of his electrical and computer engineering classmates decided to solve that problem with a senior Capstone project that puts the control in the patient's hands, or in this case, their eyes.

The undergraduate student researchers won this year's first place award in the ECE Capstone Competition for developing this eye-controlled robotic arm that allows patients to feed themselves. "Once they have the ability to do it themselves, there's an enormous sense of freedom," said James Barron, who developed software for the project.

The Capstone team included Nick Aquino, Barron, Kante, Ryan LaVoie, Pedro Lopes, and Basel Magfory. Waleed Meleis, an associate professor in the Department of Electrical and Computer Engineering, served as the group's faculty advisor.



iCRAFT has the potential "to give thousands of paralyzed individuals the independence to eat with minimal help from a caregiver," Meleis said.

Similar technologies exist — including the recently reported BrainGate implant which allows patients to control a robotic arm merely by thinking about it — but these require some kind of invasive (or even surgical) interface to connect the user's desires with the robot's behaviors, Lopes explained.

In this case, there is no physical connection between the user and the control device — no joystick under their chin, for example. Instead, the patient needs only to look at a box on a computer screen.

The team developed an eye-tracking software that couples the direction of a patient's pupils with his or her food choices. Three colored segments of the screen correspond to two bowls of food and a drinking bottle. A fourth, larger segment allows the patient to take a break from eating.

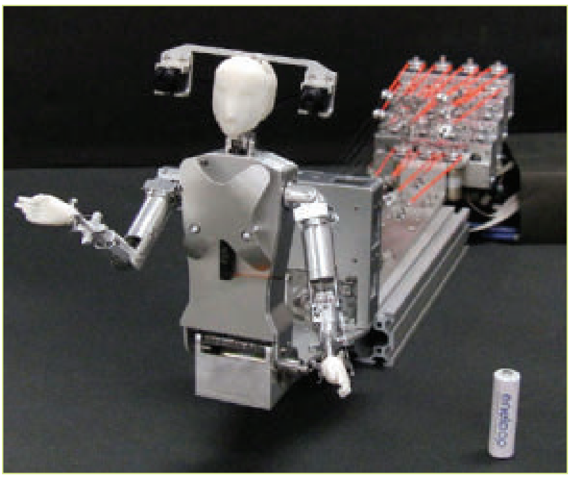
Meleis commented that the graphical user interface designed by the team is impressive because of its simplicity. The judges (12 practicing alumni engineers) "were particularly impressed with the impact iCRAFT will have on the target populations and by the successful integration of eye tracking, robotics, a custom GUI, and specialized equipment," he said.

"The single best moment of this Capstone experience was the first time we were actually able to control the robot arm with nothing but our eyes," Barron noted. "Once we were able to accomplish this feat, I was confident that everything else would fall into place."

He was right. Aside from winning first place in the Capstone Competition, the team has developed a tool that community members can use immediately with the appropriate technical know-how. The iCRAFT team has published the robotics plans online and the software package is available as an open source download as mentioned previously.

Current alternative self-feeding devices cost in the range of $3,500, but iCRAFT can be constructed for much less, making the technology a more affordable option for disabled individuals and their caregivers and families.

bots IN BRIEF

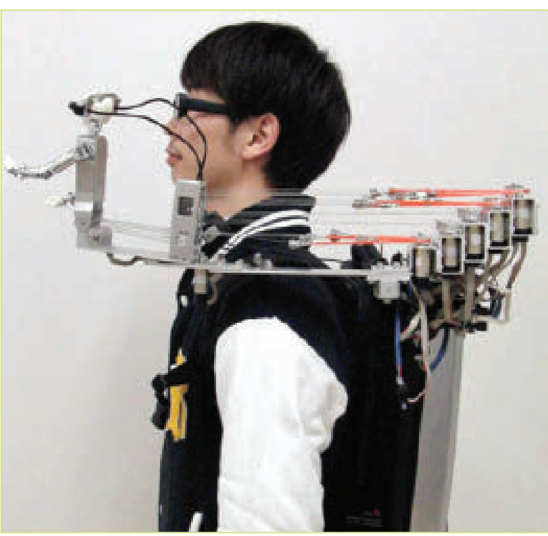


HAVE AVATAR, WILL TRAVEL

Nobody likes being alone really, and Japanese researchers from Yamagata University are developing a robot to make sure no one has to ever be alone again. The MH-2 wearable miniature humanoid lives on your shoulder and can be remotely inhabited by your friends from anywhere in the world. MH-2 (that's "MH" for "miniature humanoid") is a wearable telepresence robot that acts as an avatar for a remote operator. With two seven DOF arms, a three DOF head, and a two DOF body (plus one additional DOF for realistic breathing), MH-2 is designed to be able to mimic human actions as accurately and realistically as possible. Think Telenoid, except it can actually do stuff besides wiggle around semi-creepily.

This may seem a bit weird at first, but here's the idea: You've got a friend or a relative that you want to share an experience with. For example, you're traveling or something and you want some company. Instead of having said friend come along with you (we'll assume that they're busy as opposed to just antisocial), you can bring along an MH-2 instead. Back home, your friend puts on a 360 degree immersive 3D display and stands in front of some sort of motion capture environment (like a Kinect, for instance). Then, they get to see whatever the MH-2 sees. Meanwhile, the robot on your shoulder acts like an avatar, duplicating the speech and gestures of your friend right there for you to interact with directly. For all this to work convincingly, gestures need to be reproduced accurately and quickly at a speed equivalent to a human being making gestures in real time. This is why MH-2 is so complicated and requires that gigantic backpack full of servos which control its joints by tugging on wires.

This backpack doesn't look like a whole lot of fun to carry around for extended periods of time, which is why the researchers are trying to find ways to reduce the bulk of the 22 actuators that are currently required to operate the little guy (or girl). Until that happens, you'll just have to accept the fact that using the MH-2 could possibly make you look just a little bit like a robot geek.



MONKEY BUSINESS

This purple little guy is the Gibbot: a robot designed by the Laboratory for Intelligent Mechanical Systems at Northwestern University to explore a particular type of locomotion that's been perfected by monkeys to quickly and efficiently get around in trees.

You might remember ParkourBot — a two-dimensional gymnast robot also from Northwestern (in collaboration with Carnegie Mellon). ParkourBot is a pro at bouncing up and down walls, but it's not great at going sideways. The Gibbot, on the other hand, is designed primarily to investigate horizontal locomotion. Specifically, the Gibbot is intended to brachiate which is a type of highly efficient motion used by (big surprise here!) gibbons (which are small apes). Brachiation is essentially repetitive horizontal swinging; there's no net vertical motion which means that the gibbon doesn't really have to expend much in the way of energy fighting gravity. Once it gets going, the gibbon can move very fast by just grabbing on and letting go at the right times. Figuring out what these times are (and what gaits they result in) is the tricky part, but the researchers were able to show off some successes. The Gibbot itself consists of two arms with electromagnets at the ends and one powered joint in the middle. It swings around on a steel wall which provides an unlimited number of clamping points for the magnets. This allows for the testing and comparison of a variety of different brachiating gaits, with a fairly ambitious goal in mind, according to their paper: "By employing a diverse suite of gaits, the Gibbot will be able to perform gymnastic maneuvers to reach specific handholds in the environment."

DASH TO THE FINISH

DASH — UC Berkeley's 10 centimeter long, 16 gram Dynamic Autonomous Sprawled Hexapod — has learned a new trick. The robot can now perform "rapid inversion" maneuvers, dashing up to a ledge and then swinging itself around to end up underneath the ledge and upside-down. This replicates behaviors in cockroaches and geckos, and may lead to a new generation of acrobatically-inclined insectobots.

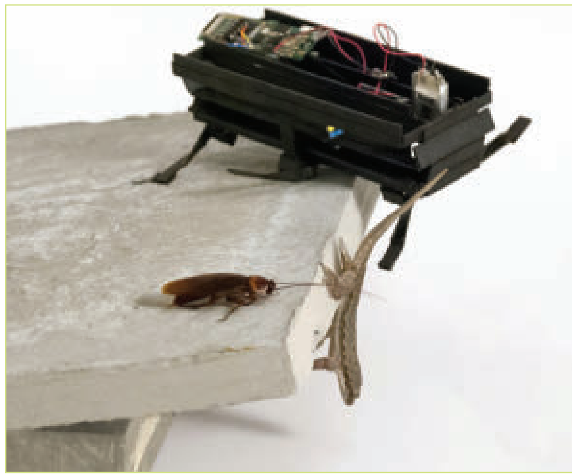


Photo: Jean-Michel Mongeau, Ardian Jusufi, and Pauline Jennings (UC Berkeley PolyPEDAL Lab)

Cockroaches have a notorious ability to vanish from sight before your brain even decides you should take a swat at it, and if you've ever tried to chase down a gecko, you know that they're not just fast, but they're also incredibly agile. These abilities stem in great part from the fact that cockroaches and geckos are small and light, and consequently don't have to overcome much inertia when changing direction. We've only recently been able to take advantage of technologies that allow for the creation of robots at similar scales, and such robots (like DASH) exhibit impressive speed and agility.

Recently, researchers at UC Berkeley's PolyPEDAL Lab (led by Professor Robert Full) demonstrated that cockroaches can perform "rapid inversions" on a ledge — a previously unknown behavior. Surprisingly, while on a vacation research trip at the Wildlife Reserves near Singapore, the researchers discovered a similar behavior in lizards, and documented geckos using this technique in the jungle to escape predators and nosy scientists. Full's group then teamed up with roboticists from Berkeley's Biomimetic Millisystems Lab to see if DASH could be taught to do the same sort of thing.

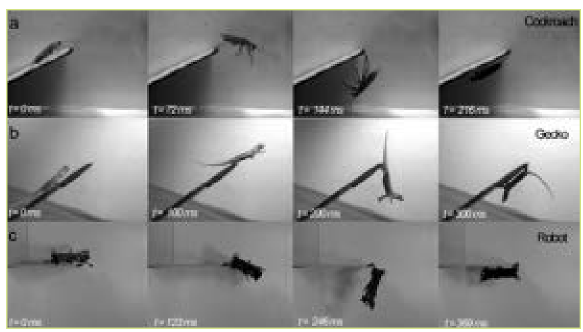
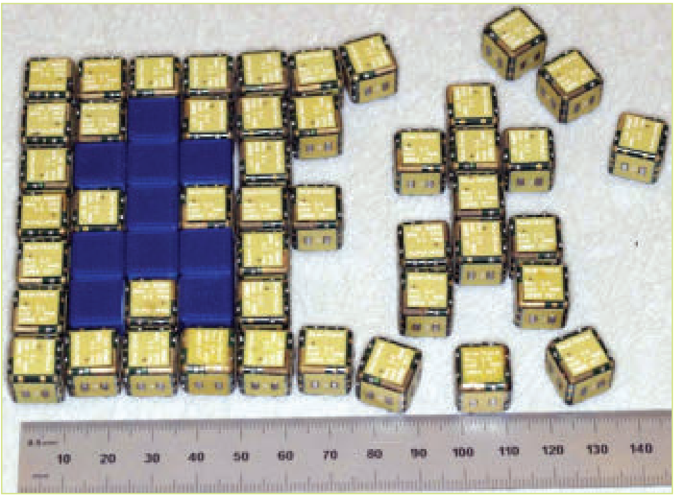


Image: PLoS One

DASH — unlike cockroaches or geckos — doesn't come with claws, so the researchers "simulated claw action" by sticking some Velcro™ onto DASH's front and hind legs, and then adding more Velcro to the top and underside of the ledge to form pivot and catch points. Since the whole Velcro thing is kind of cheating if you're trying to design a robot inspired by animals (as opposed to plants), the Berkeley researchers have started to develop designs for both active and passive bio-inspired claws. With the ability to naturally stick to surfaces and perform these new acrobatic tricks, the UC Berkeley teams say DASH could soon be able to make speedy transitions between running and climbing, eventually leading to "highly mobile sentinel and search-and-rescue robots that assist us during natural and human-made disasters."

Note that no cockroaches or geckos were harmed over the course of this research.



SHIFTY SAND

Imagine that you could toss an object such as a wrench into a container filled with tiny robots and — within seconds — the robots would "sense" the shape of the wrench and bind to each other to form a replica of the tool. Creating robots that could turn this sci-fi-like scenario into reality is the goal of an MIT team led by Professor Daniela Rus. They call the technology Smart Sand.

The project still has a long way to go. The robots the researchers have developed consist of relatively large cubes (each 12 millimeters on a side), but the team hopes to be able to shrink the modules in the future. In the meantime, the group is addressing another challenge: how to convey the shape of an object to lots of modules that have limited computational resources.

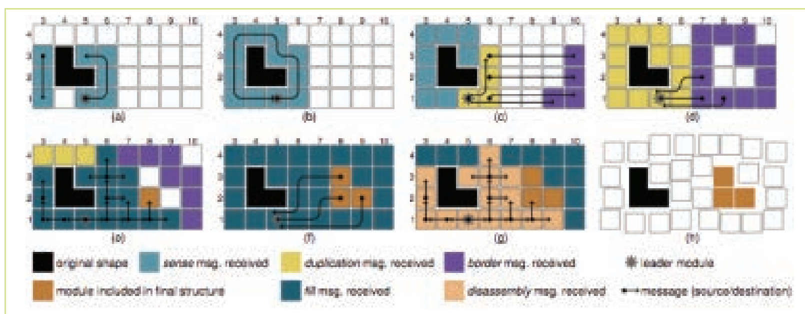
At this year's IEEE International Conference on Robotics and Automation (ICRA) in St Paul, MN, Kyle Gilpin (one of the researchers) presented an algorithm capable of doing just that. He also showed video of a working prototype that — although it can duplicate only a very simple object — shows the potential of the technology.

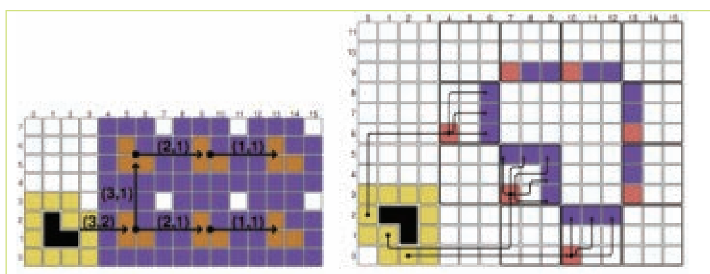
Gilpin explained that each module — or smart pebble — has only a small amount of processing power and memory, so individual modules cannot store a complete digital map of the object to be assembled. Each has to acquire only a certain amount of information, and by sharing pieces of data they must solve the problem collectively.

In the algorithm Gilpin and Rus came up with, an ensemble of modules first detects the border of the object. This ensemble then sends messages to another group of modules which recreate the border pattern. Finally, these modules within the border attach to each other, whereas all other modules self-disassemble — leaving only the original and the replica. The illustration here shows the method step by step.

The researchers tested their algorithm hundreds of times using different shapes. They say the algorithm can handle communication failures among modules, and it can create multiple replicas of a single original or create bigger replicas. They also claim that the technique works efficiently for 3D objects, as well (by slicing the object into 2D layers).

Shrinking the cubes remains the biggest challenge. Gilpin believes that they could eventually achieve dimensions of about one millimeter, in effect transforming their smart pebbles into smart sand, but that will take time. One may wonder why we need smart sand in the first place when technologies like 3D printing already allow people to easily create replicas of objects.

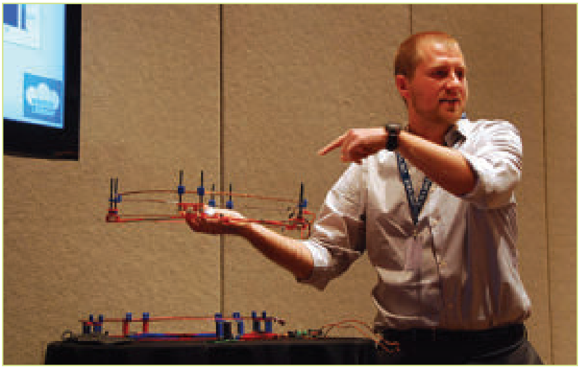




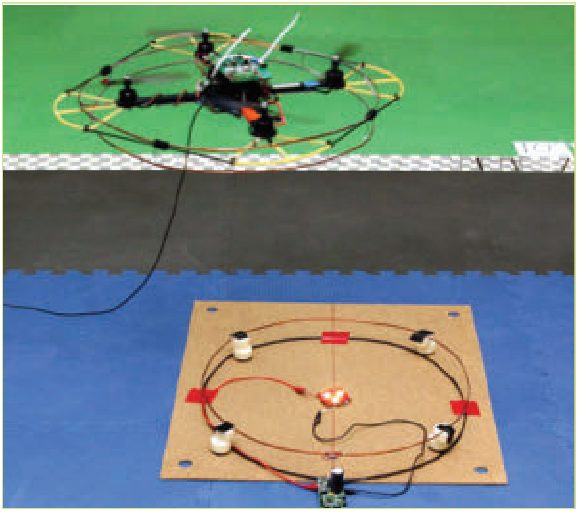
Gilpin says one advantage is that objects built with smart sand could have sensing and processing capabilities. For example, you could build an electronic torque wrench. What's more, smart sand based objects are able to recycle themselves. So, when you're done using your wrench, it would dissolve and turn into, say, a screwdriver. That's a futurist scenario, of course, but Gilpin is excited about the possibilities. "You don't need to carry all your tools with you. Just a bag of sand."

Cool tidbits herein provided by www.botjunkie.com, www.robotsnob.com, www.plasticpals.com, http://www.robots-dreams.com/, and other places.

WHAT'S THE CHARGE?



Transmitter and receiver coils.



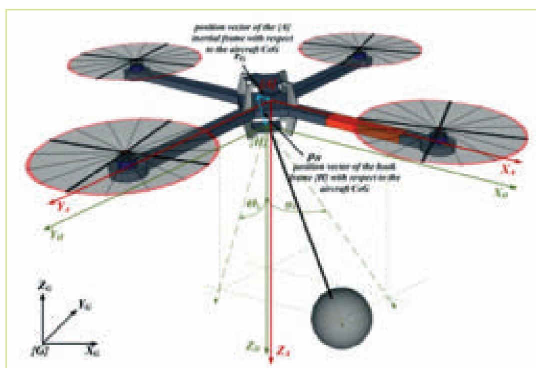
UAV wirelessly transferring power to light a LED.

Whenever you have something that needs to survive away from the electrical grid for an extended period of time, batteries are inevitably the limiting factor, and roboticists from the NIMBUS Lab at the University of Nebraska-Lincoln have solved the problem with a quadrotor that can fly around and wirelessly charge up electronics for you. The type of wireless power that these quadrotors are beaming out is based on what's called "strongly coupled magnetic resonances." Basically, you've got two coils of wire: one on the quadrotor, and one on whatever you want to power or charge (we'll call this the receiver). The quadrotor drives a current in its coil which generates an oscillating magnetic field. When the quadrotor gets close enough to the receiver, the receiver's coil starts to resonate with the magnetic field transmitted by the quadrotor. That resonance induces a voltage in the coil which the receiver can use to power its electronics or charge its battery.

Exactly how much power gets from the quadrotor to the receiver (and how efficient the transmission is) depends largely on how well the quadrotor can keep close to the optimum transmission distance, which is about 20 centimeters away from the receiver coil. When everything works perfectly, the quadrotor can wirelessly transfer about 5.5 watts of power with an efficiency of 35 percent — which is easily enough to power a light. This does, of course, decrease the flight time of the quadrotor, but the whole point is that it can just fly back to base and recharge itself. Stationary electronics don't have that option.

As far as applications go, the researchers suggest that this kind of system would be great for "highway messaging systems, ecological sensors located in forests, or sensors shallowly embedded underground or in concrete." UAVs would act as mobile power stations, zipping around and delivering power to sensors when necessary. Rumor has it that CyPhy Works (the stealthy startup run by iRobot co-founder Helen Greiner) is developing UAVs for infrastructure inspection, and it seems like some wireless sensor charging capability would fit right in with that sort of thing. The researchers are staying busy teaching their quadrotor to autonomously keep a stable hover at an optimum distance to transfer power to a receiver, which will involve either using something like a camera or feedback from the power transmission itself. They're also hoping to be able to boost the amount of power the quadrotor can transfer, and maybe toss in some super capacitors.

GETTING INTO THE SWING OF THINGS



One of the ways in which robots are starting to get really useful is with hauling aerial cargo. Last year, the optionally-manned KMAX made its first autonomous cargo delivery in Afghanistan, and since it can fly as many missions as you have fuel to keep it going, it's definitely a safer and more efficient way to get supplies to troops — especially in dangerous areas.

To move cargo around, helicopters (autonomous or otherwise) often carry stuff slung beneath them on long ropes. As you can probably imagine, said cargo often ends up doing all sorts of swinging about, especially if the helicopter that's carrying it has to maneuver. Researchers from the University of New Mexico have been developing algorithms that allow robots to compensate for motion-induced swinging of suspended loads, and are testing them out on real live quadrotors.

Essentially, what the quadrotor is doing here is dynamically adjusting its trajectory to damp out the swinging motion of its cargo. It's sort of like an upside-down version of pendulum balancing, with maybe a little bit of hinged stick balancing thrown in for good measure. Next, the researchers plan to see if they can get their algorithms to work on platforms that are less balanced (and more realistic) which (they say) should be "an important step towards developing the next generation of autonomous aerial vehicles."

Firgelli
www.firgelli.com
LINEAR SERVOS
L12-R Linear Servo
• Direct replacement for regular rotary servos
• Standard 3 wire connectors
• Compatible with most R/C receivers
• 1-2ms PWM control signal, 6v power
• 1", 2" and 4" strokes
• 3-10 lbs. force range
• 1/4" to 1" per second speed ranges
• Compatible with VEX
L16 Linear Actuators
• 2", 4" and 6" strokes
• 10 - 40 lbs. force range
• 1/2" to 1" per second speed ranges
• Options include Limit Switches and Position Feedback
New!
PQ12 Linear Actuator
• Miniature Linear Motion Devices
• 6 or 12 volts, 3/4" stroke
• Up to 5 lbs. force
• Integrated position feedback or limit switches at end of stroke
• External position control available
Linear Actuator Controller (LAC)
• Will drive any Linear Actuator with position feedback
• Up to 24v and 4 Amps
• USB connectivity to drive the actuaor with your computer
• Adjustable speed, limits and sensitivity
L12-NXT Linear Servo
• Designed for LEGO Mindstorms NXT®
• Plugs directly into your NXT Brick
• NXT-G Block available for download
• Can be used with Technic and PF
• Max. speed: 1/2" per sec.
• Pushes up to 5 lbs.
• 2" and 4" strokes
Available Now @
www.firgelli.com

CanaKit™
www.canakit.com
$149.95
New
USB-LOGIC
USB Logic Analyzer
$99.95
DSO-NANO-V2
Pocket-Sized Digital Oscilloscope
$59.95
ARDUINO-STK
Arduino Uno Starter Kit
$59.95
ASSEMBLED
UK1104
4-Port USB Relay Controller with 6-Channel Analog/Digital Interface
FM Transmitters • Timers • Audio Amplifiers • Motor Controllers • Custom Designs
www.canakit.com
1•888•540-KITS (5487)

EARN MORE MONEY
Get your dream job!
Be an FCC Licensed Wireless Technician!
Make up to $100,000 a year and more with NO college degree
Learn Wireless Communications and get your "FCC Commercial License" with our proven Home-Study Course!
• No need to quit your job or go to school.
• This course is easy, fast and low cost.
• No previous experience needed!
• Learn at home in your spare time!
Move to the front of the employment line in Radio-TV, Communications, Avionics, Radar, Maritime and more… even start your own business!
Call now for FREE info
800-877-8433
ext. 210
Or, email us:
fcc@CommandProductions.com
COMMAND PRODUCTIONS
Warren Weagant's FCC License Training
P.O. Box 3000, Dept. 210 • Sausalito, CA 94966
Please rush FREE info immediately!
NAME:
ADDRESS:
CITY/STATE/ZIP:
email: fcc@CommandProductions.com

COMBAT ZONE

Discuss this section in the *SERVO Magazine* forums at http://forum.servomagazine.com

Featured This Month:

BUILD REPORT:

Siafu: An Army of Ants – Part 3

● by Pete Smith

In Part 2 of this series, I completed the design process for my new Antweight drum bot. In this part, I will show how the parts were manufactured, provide a list of the other parts used, and how they went together to make a drivable chassis.

The various body panels were to be water cut. I created a dxf file of each part (**Figure 1**). Remember to remove any features from the model that will not be included in the cut; in my part, that was the holes for mounting the top and bottom panel. I added a 1" square to the drawing so that the manufacturer would know the correct dimensions of the part.

I decided to order 10 of each part since I was pretty confident in the design, and the cost of 10 sets of parts was not that much more than the cost for one or two. The parts were ordered from Westar (**www.westarmfg.com**), the water jet company associated with Team Whyachi.

The parts soon arrived (**Figure 2**). I ordered the required screws from **www.mcmaster.com** and cut the required sections of nutstrip from what I had in stock.

I fitted mounting plates to two of the motors (**Figure 3**). The thicker plate worked well, and standard 3 mm long screws

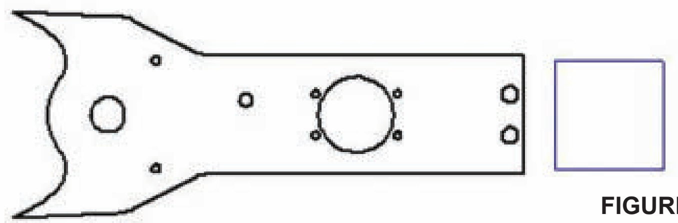


FIGURE 1.

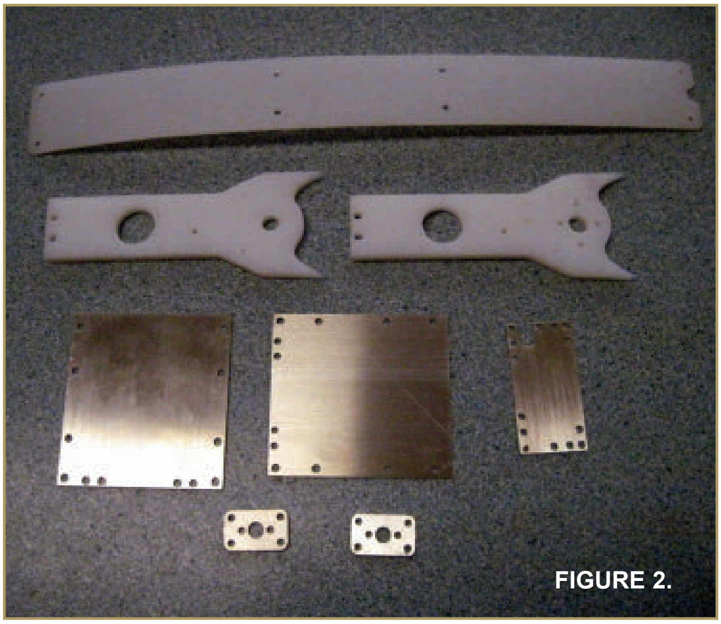


FIGURE 2.

FIGURE 3.

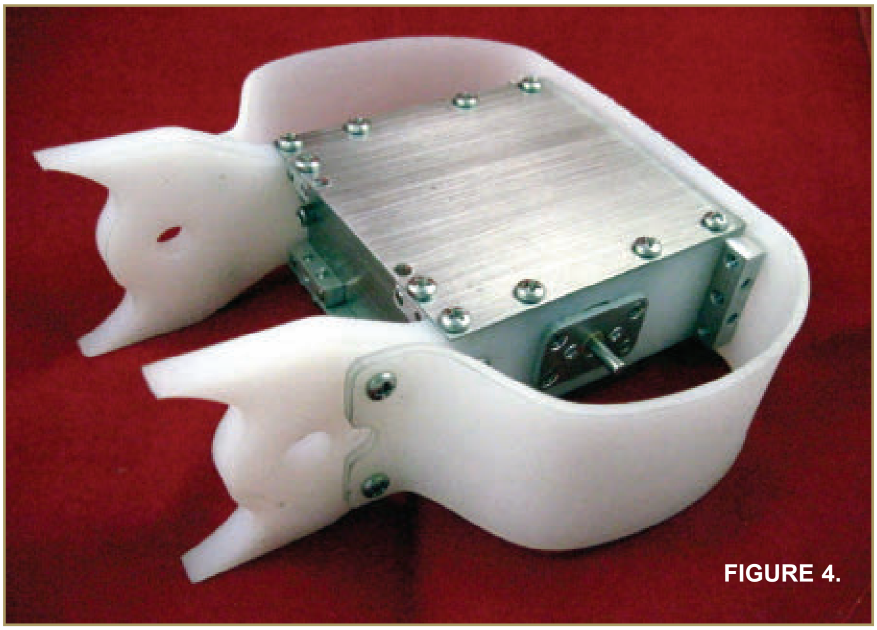


FIGURE 4.

secure the plate without interfering with the gears.

A few holes in the chassis plates were a little too small to clear the 4-40 screws used, but once these were drilled out the chassis came together easily (**Figure 4**). It is as robust as I could have wished for and should prove tough enough for combat.

I had previously purchased most of the other parts I intended to use in the design. This allowed me to create 3D models of the parts and also get accurate weights to put in a spreadsheet to make sure the design did not go overweight.

The weapon speed controller is a Turnigy Trackstar 18A TTS-18A (**Figure 5**); this has excessively long leads and an unnecessary on/off switch. I trimmed the leads, shorted out the switch leads so it is always "on," and replaced the signal leads with those suitable for an Orange Rx 415 micro receiver. This reduced the weight from 0.9 oz to 0.7 oz, and more importantly, made it a lot more compact (**Figure 6**).

This controller is easily programmed using its matching program card, but to date I can only get it to run at 50% when in reverse. I believe this can be manually programmed using the TX and various beeps from the motor, but I've not tried to do that yet. Reverse is useful if the bot gets inverted.

Weta uses a 850 mAH 3S LiPo, so I thought that a Turnigy nano-tech 300 mAH 2S would probably suffice for the Ant, but there will be room and weight for a bigger pack if required.

The drive's ESCs will be either Fingertech (**www.fingertech robotics.com**) tinyESC v2 or Botbitz (**www.botbitz.com**) ESCheap6 style. Actually, for the prototype it will be one of each because I blew the second of each set by plugging a battery in the wrong way. JST connectors may be polarized, but not well enough to prevent accidental reversal if you are not paying attention. Both were fitted with the signal cables to match the Micro RX and covered in new heat shrink (**Figure 7**). The tinyESC v2 is smaller and lighter and has an excellent reputation, but the ESCheap6 is — as the name suggests — a little cheaper.

I added the drive ESC, RX, and

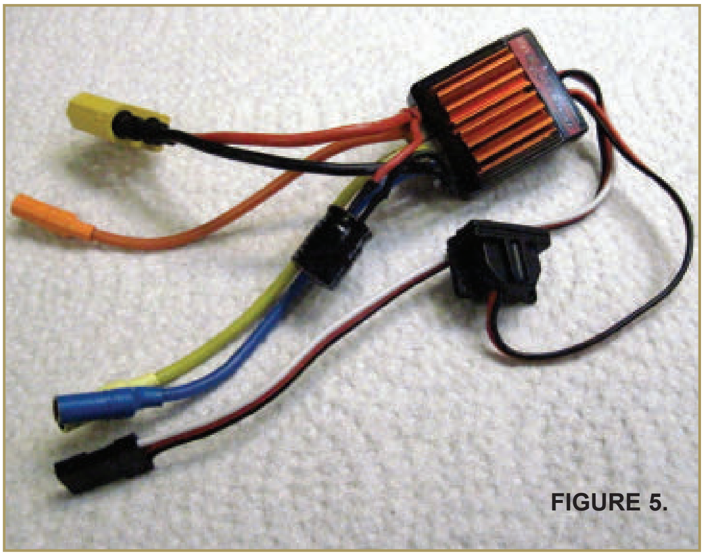


FIGURE 5.

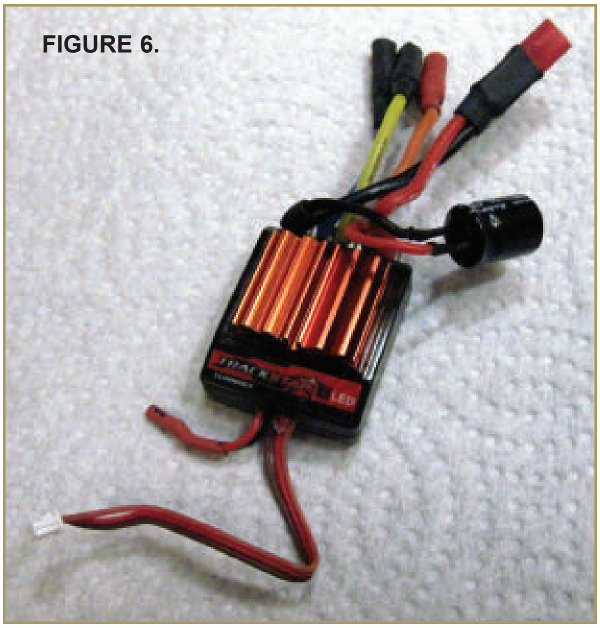


FIGURE 6.

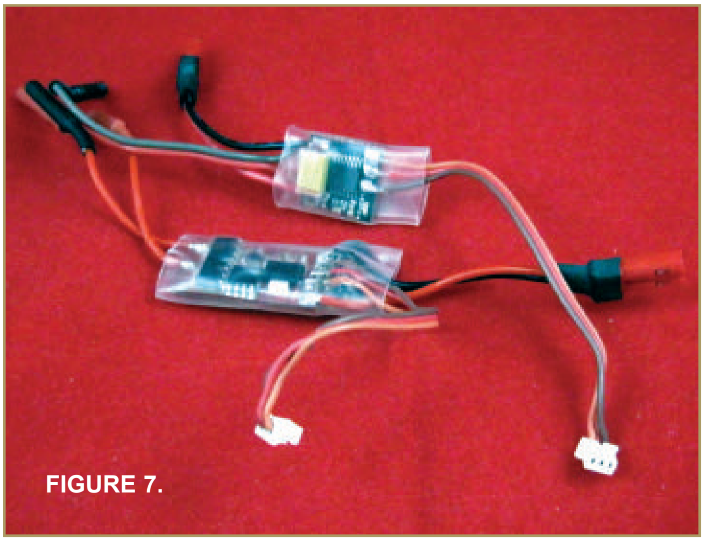


FIGURE 7.

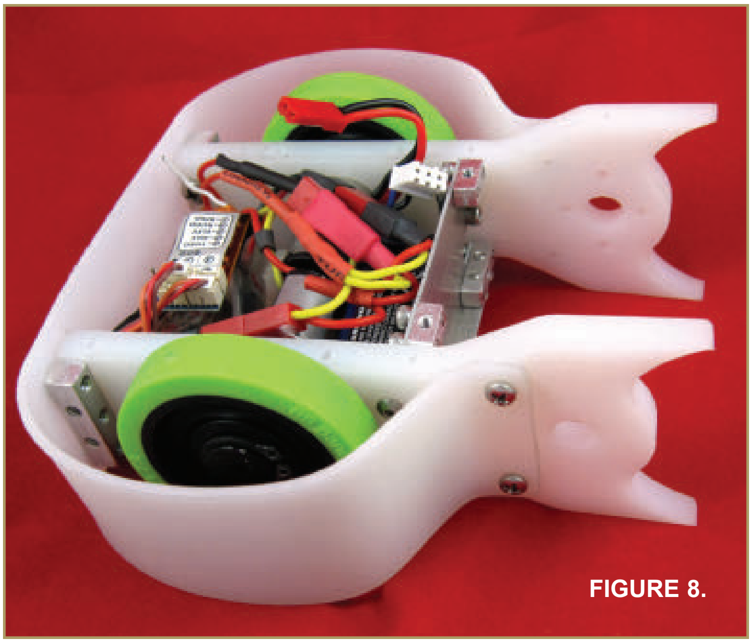


FIGURE 8.

battery to the chassis (**Figure 8**) and gave it its first test run. Performance was as expected and video of the first run can be seen on YouTube by searching "Saifu first run"

Part 4 will describe how I made the drum for the bot, and completed the rest of the build. **SV**

EVENTS

Recent Events

Winter Challenge 8 was held by RoboCore at the Faculdade de Jaguariúna, Brazil, June 7-10th.

Upcoming Events

Australian Robowars Nationals 2012 will be presented by the Queensland Robotic Sports Club in Brisbane, Queensland, Australia, September 29th through October 1st. **SV**

The History of Robot Combat: Motorama

● by Morgan Berry

Motorama was born around the same time as many other robot combat events: shortly after the end of BattleBots. Through the years, the competition has grown considerably, changed arenas, seen child competitors grow up in the sport, and evolved.

One thing that remains the same about the event is the atmosphere of camaraderie and friendship that is unique to all robot combat circles.

NERC (North East Robotics Club) was founded in 2000, and is one of the longest running robot combat organizations currently working in the United States. In addition to hosting Motorama every year, they also host an event in conjunction with the Franklin Institute Science Museum in Philadelphia.

NERC uses its own rules, and typically organizes events in a double elimination style. Their goal is to "make robot combat accessible to all interested parties." The organization prides itself on being a "small, close-knit family of like-minded builders," and is working to keep the spirit of robot combat alive and well in the northeast.

The Motorama event has weight classes spanning from 150 grams (Fairyweight) to 30 pounds. In addition to awarding first, second, and third place prizes in all of these categories, the judges also give out awards for "The Coolest Robot," "The Most Destructive Robot," and "Best Driver."

The robot combat event at Motorama is part of a large motor show held in Harrisburg, PA every year. Robot combat is a relatively new addition to the event; Motorama has been around since 1978.

In addition to combat, visitors to Motorama can view a massive show of antique and modern automobiles and numerous indoor racing events. This year was a milestone for the Motorama robot combat competition; builders celebrated the 10th anniversary of its founding, making it one of the longest running competitions in the sport. In honor of the occasion, *SERVO* spoke with Al Kindle — long-standing NERC officer and Motorama attendee — about the past, present, and future of the event.

SERVO: What was the first Motorama competition like?

Al Kindle: It was held in 2003 with the old NERC 16 ft arena. I want to say we had over 100 bots as all Motorama events that I can recall have had 100+. The Builders Database does not go back as far as this event.

SERVO: Has a lot changed over the years at Motorama? How has the competition evolved over time?

Al: The arena has changed several times. In 2004, we used the 24 ft Toad Tank, graciously donated to NERC by Michael (Fuzzy) Maulden and the Lazy Toad Grill. We ran bots from Antweight through Lightweight at this Motorama. Roy Helen was on hand for this as MC in his full BattleBot armor. Roy would return for several other Motorama events; the Toad Tank would not. The costs involved with keeping and transporting the Toad Tank were beyond NERC's capabilities. Since then, we have used variations of the POP 12 ft arena provided by Robert Masek. The current arena is NERC's own custom built 16 ft arena, designed and built by Robert Masek and Eric Scott. The arena is stored in a barn on the property of the awesome Benson family, who also transport the arena to and from the Motorama venue back to its home in Winchendon, MA. We could not have Motorama events without their tremendous dedication and generosity.

SERVO: What age groups participate at Motorama? Is the body of competitors getting younger/older/staying the same over time? Does NERC do anything to reach out to young people at Motorama?

Al: We have had competitors from as young as four all the way into their 80s. This truly is an ageless hobby. We have seen young competitors age and remain active with us as new young people become involved. We have seen an increased involvement with area schools, as well. Georgetown Day School Robotics, for example, out of Washington DC.

We recently began running the Bots IQ 15 lb weight classes at Motorama and have had an encouraging number of competitors in the class.

SERVO: How many participants would a Motorama event typically have?

Al: We routinely have 100-125 bots at each Motorama event.

SERVO: How far away do most

people come for the event?

Al: We have regular competitors from as far away as Colorado and Canada. We also have builders from many different states: New Jersey, Pennsylvania, Florida, Texas, and North Carolina, just to name a few.

SERVO: What do you think the future holds for Motorama and robot combat?

Al: 2012 was our 10th event at Motorama and we hope to have at least another 10. We also have our annual event at The Franklin Institute in Philadelphia in October. We will do our part to continue the growth and advancement of the combat robotics hobby in the northeast as long as it is feasibly possible.

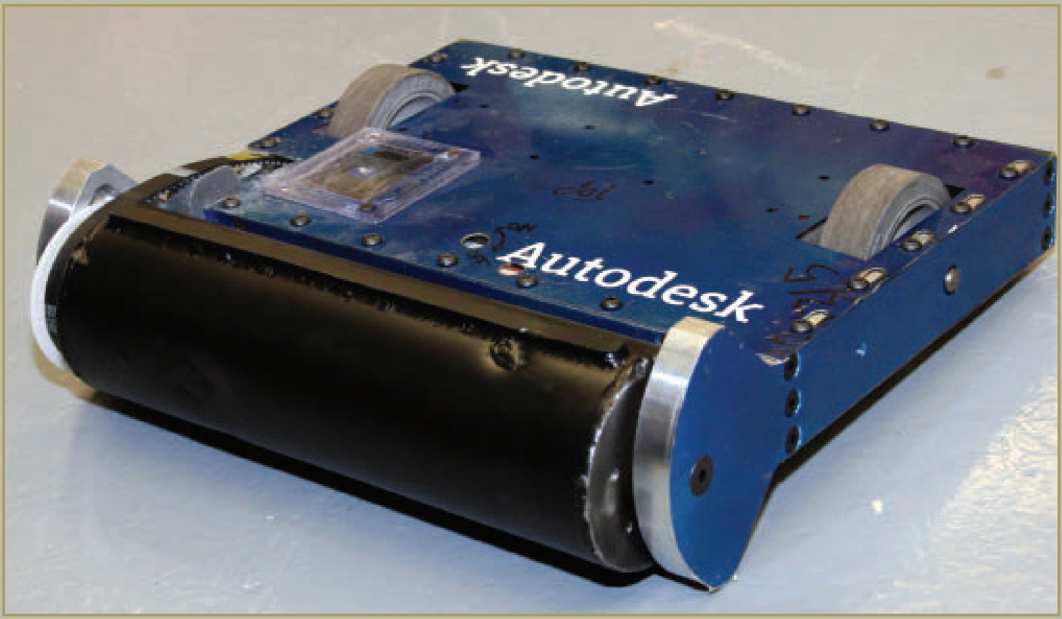
Drawing over 100 bots every year to Harrisburg, Motorama is one of the largest currently running events in America. Aside from just plain being fun, Motorama also does a good job of drawing in new young faces into the sport, and keeping them involved all through their youth. This is an important task, since keeping the youth involved in events is crucial for the sport to survive and thrive.

In the **photos**, you'll find a few of the competitors at Motorama this year from various weight classes. **SV**

www.nerc.us

Hedgehog — a Fairyweight — has a spinning blade and was entered by Team Mateo.

Tetanus is a 30 lb spinning shell bot from Team Brain Damage. It won first place in the Featherweight class at Motorama this year. It also took home the coveted "Most Destructive" award.



Higgins from Team Tech uses a drum spinner for a weapon.

From Near Chaos Robotics, one pound Kobalos is a horned wedge design. It took home second place in the Antweight category this year.

Another entry from Near Chaos Robotics was Featherweight Moros, using a horizontal spinner weapon.

Bring Back the Pogs

● by Dave Graham

I've been involved in combat robotics for a number of years as both a competitor and an event organizer. One of the things I really enjoy about the sport is the variety of mementos created by the innovative minds of robot enthusiasts. I attend many events and normally compete in the Insect class of fighting robots.

Cracking the top three at a competition can garner you a reward ranging from nothing to a custom designed trophy. While I've enjoyed my share of top three finishes, many times that was not the case and I came away with nothing. As a result, I really liked the competitions that offered pogs to match winners. Even if I didn't make the top three, the odds were I'd win a match or two along the way and pocket a few pogs.

My first win at a big competition was February 2007 at the NorthEast Robotics Club's (NERC) Motorama. The match pog was a simple metal disc stamped with "NERC WINNER" that looked more like something you'd find on a dog's collar (**Figure 1**). I won my second pog in April 2007 at the Upstate New York Robot Battle (**Figure 2**). The competition was held at the Carousel Mall in Syracuse, NY (**Figure 3**), and as it turned out it was the last year the event was held. The match winner pog is the only memento I have of that event.

Pog three looked more like a poker chip and was added to the collection in July 2009 thanks to Jeremy Campbell and his BotBlast in Bloomsburg, PA (**Figure 4**). NERC was back in the pog business in February 2010, this time awarding plastic saw blade pogs etched with their NERC logo and Motorama 2010 (**Figure 5**). My local fighting robot club, Pennbots in Boiling Springs, PA began issuing match winner pogs about the same time (**Figure 6**).

The last addition to my pog collection came from Dave Wiley who runs the California Insect Bots competitions. I met Dave at RoboGames this year, and eventually the conversation got around to match winner pogs. Dave offered me a set of the Insect class match winner pogs he used for his Bot Gauntlet event (**Figure 7**).

FIGURE 1. NorthEast Robotics Club (NERC) match winner pog, February 2007.

FIGURE 2. Upstate New York Robot Battle (UNYRB) match winner pog, April 2007.

FIGURE 3. UNYRB at the Carousel Mall in Syracuse, NY in April 2007.

FIGURE 4. BotBlast match winner pog, July 2009-2010.

FIGURE 5. NERC match winner pog, February 2010.

The pogs are actually wooden ice cream spoons Dave stamped with either a Beetle, Ant, or Flea symbol (remember the innovative minds!). If you look closely at the Flea pog, you'll notice it has what appears to be a plastic eye glued to it. Dave assured me it was not a plastic eye. He claims it is a small robotic flea in a glass case (**Figure 8**).

I really enjoy and cherish my pog collection and hope that more event organizers will start awarding match winner pogs. SV

FIGURE 6. PennBots match winner pog, 2010-present.

FIGURE 7. California Insect Bots Bot Gauntlet match winner pogs: Beetle (left); Ant (center); and Flea (right).

FIGURE 8. Close-up of California Insect Bots Bot Gauntlet Flea match winner pog.

People of Bots

Puerto Rican Pride at the South Florida Robot Riot

● by Dave Graham

Puerto Rican pride was in the house at the South Florida (SFL) Robot Riot as students in grades 7-12 from the Central Visual Arts School in San Juan, Puerto Rico made their presence known through both their music and fighting robot skills. The SFL Robot Riot is an open Insect class fighting robot event conducted in conjunction with the STEM Tech Olympiad 2012 at the Miami Convention Center the last weekend in April.

Two Puerto Rican teams — RoboCAV in yellow shirts and Abusement Park in green shirts (**Figure 1**) — made the journey from San Juan to Miami to compete in the three day fighting robot event, along with their six three pound Beetleweight bots. Throughout the competition, the Puerto Rican students proudly displayed their musical heritage by breaking into song to the beat of their tambourines.

One morning, they sang as they paraded from the Convention Center entrance all the way to their pit area. CNN Espanol interviewed several members of team RoboCAV (**Figure 2**) who explained the fighting robot weight classes and double-elimination format to the reporter. The students ended the segment by entertaining the CNN reporting team with a sample of their music (**Figure 3**).

Team Abusement Park had the final word as their three pound Beetleweight bot Phineas beat one of the best (if not the best) Beetleweight bots and drivers in the country — Gene Burbeck and his bot One Fierce Lawnboy — to take the gold. Members of the team display Phineas, their first place trophy, and a gift certificate donated from the Robot Marketplace (**Figure 4**). SV

FIGURE 1. Puerto Rican teams RoboCAV in yellow shirts and Abusement Park in green shirts.

FIGURE 2. CNN Espanol interviewing members of team RoboCAV.

FIGURE 3. Members of team RoboCAV singing for the CNN Espanol reporting team.

FIGURE 4. Members of team Abusement Park with their first place robot Phineas.

People of Bots

Meet the Robot Riot Teams

● by Andrea Suarez

The first annual Robot Riot attracted 38 robots to Miami Beach from Puerto Rico, Minnesota, Michigan, New York, Pennsylvania, and Florida. Competitors ranged from 13-60 years of age, including five teams that fought their first robot at this event. Among these new builders was Hugh Savoldelli, who surprised everyone by winning the Antweight division undefeated! Here's what the teams had to say:

Why do you build robots?

"I can't help it! Seriously, watching a creation come "to life" from idea to completion." — *Jim Smentowski, Team Nightmare (Florida).*

"I love the design challenge and the skill aspect of driving." — *Eric Mueller, Team Hazardous Robotics (Minnesota).*

"Fun to see robot carnage, preferably the opponent's!" — *Win Halelamien, Ransom Everglades (Florida).*

"Because it's pretty exciting, that feeling of competence. What I like most is to learn new things and how these mechanisms could be used for many things in our world." — *Onassis Jr. Romero, Team RoboCAV (Puerto Rico).*

"Building robots allows me to release the pent-up "artist" in me in a creative and exciting medium, while also providing a social break and creating new avenues for me to learn. Besides, it's a cool resume add-on." — *Paul Grata, Busted Nuts Robotics (Florida).*

How did you start building robots?

"After seeing Robot Wars live in 1996, I was hooked. Built my first robot for '97 Robot Wars. Got destroyed by Jamie Hyneman. Decided I had to beat him so I came up with Nightmare, but never got to fight him." — *Jim Smentowski, Team Nightmare (Florida).*

"I started when I saw someone building their 120 lb walker at an R/C car race when I was 12. I have been building since I was 15." — *Eric Mueller, Team Hazardous Robotics (Minnesota).*

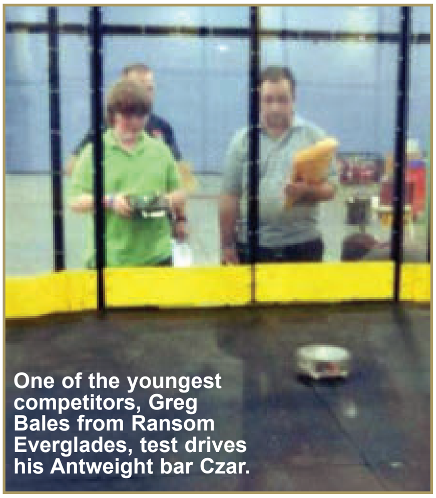
"I built my first robot when I was nine years old and competed at NERC/ Motorama in PA. I also built small combat robots in grades 4 and 5." — *Greg Bales, Ransom Everglades (Florida).*

"I've always liked to build machines and take electronics apart since I was very young (four years old). I'm homeschooled. I took private classes with an engineer for two years and learned a lot about and built a Boe-Bot. Last fall (2011), I started taking classes with Bill Garcia at StarBot. This is my first competition." — *Adrian Botran, Team Adrian (Florida).*

"I started as part of a college extra-curriculum activity. Participated in various BotsIQ competitions with Team Metal Twisters 120 lb Rhino. After graduation, I liked it so much

Team RoboCAV's toolbox has paintbrushes next to their screw drivers! (*Photo by Jaime Balzac*).

Builder's dinner at "5 Napkin Burger" on South Beach's Lincoln Road.



One of the youngest competitors, Greg Bales from Ransom Everglades, test drives his Antweight bar Czar.

that I had an idea with my friend Gregg Williamson to create a high school team. So far, we have been taking kids from that high school for two years and had a bunch of fun." — *Jaime Balzac, Team Davincinators (Puerto Rico).*

Where do you build your robots?

"Bedroom assembly." — *Eric Mueller, Team Hazardous Robotics (Minnesota).*

"We build the robots at a local racing machine shop." — *Jaime Balzac, Team Davincinators (Puerto Rico).*

"Garage-built." — *Jim Smentowski, Team Nightmare (Florida).*

"Basement." — *Dave Graham, Team Mateo (Pennsylvania).*

What's your background? What sets your team apart?

"Since it is a team composed of kids from a high school specialized in visual arts, they made a mutation of DaVinci and Terminator." — *Jaime Balzac, Team Davincinators (Puerto Rico).*

"Currently owner of Robot Marketplace. Past included doing movie special effects at ILM (George Lucas)." — *Jim Smentowski, Team Nightmare (Florida).*

"43 years federal service with the army." — *Dave Graham, Team Mateo (Pennsylvania).*

"We build really cool robots. The time we put forth into a design goes beyond meticulous, straight into the smash brothers and pizza phase of brain-dead early morning chaos." — *Paul Grata, Busted Nuts Robotics (Florida).*

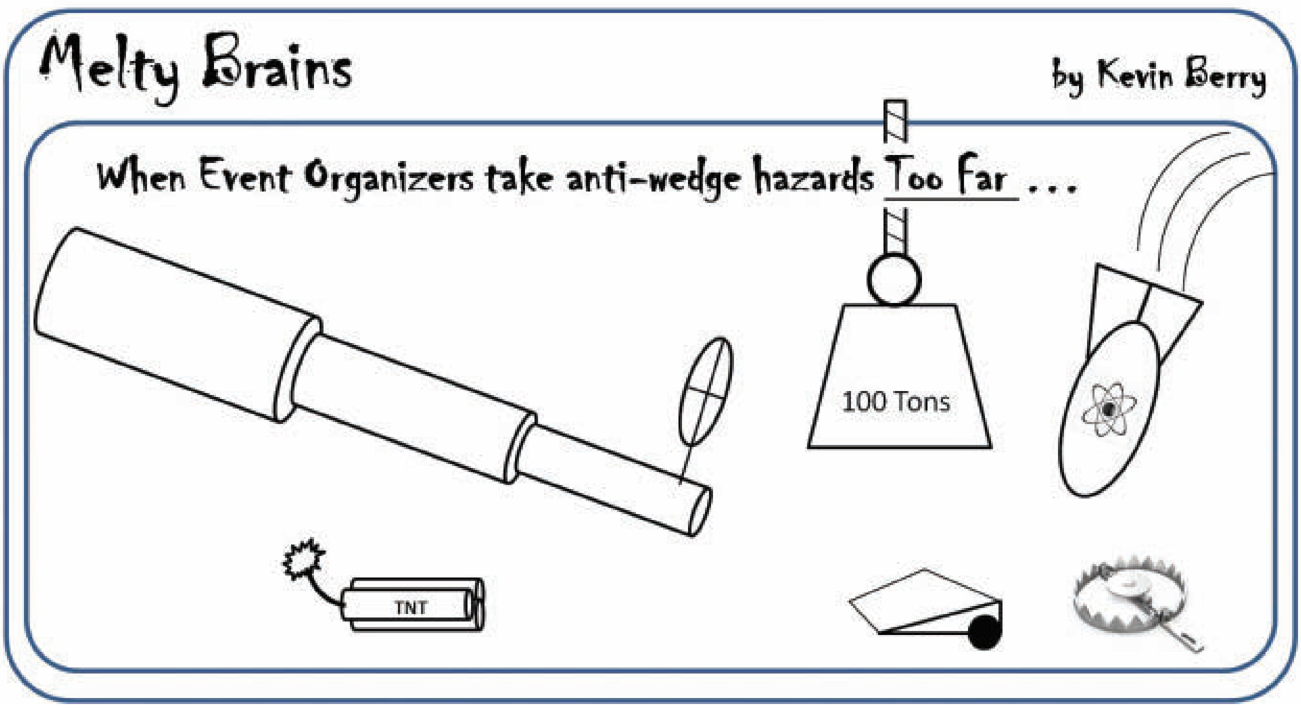
"Electronics. Controls engineer at Guardian MFG. I build ozone machines." — *Stan Marion, Team BotWorks (Florida).*

"Light saber building, Boy Scouts, and inventing." — *Greg Bales, Ransom Everglades (Florida).*

"The three of us are from research and development engineers for medical device companies. Fighting robots helped us understand how things break, how to fix them, and how to design things that don't break — critical for our jobs!" — *Andrea Suarez, Busted Nuts Robotics (Florida).*

"Currently, I'm interested in making knives and blacksmithing. I've made three knives from recycled materials and created my own forge in my backyard from scratch." — *Adrian Botran, Team Adrian (Florida).* SV

Jose Santos (Davincinators) and Win Halelamien (Ransom Everglades) take some time between matches to enjoy the company of fellow robot builders. (*Photo by Jaime Balzac*).



People of Bots

Did Jim Smentowski Pass the Torch?

● by Dave Graham

BattleBots legend Jim Smentowski attended the South Florida (SFL) Robot Riot in Miami Beach the last weekend in April as both a competitor and an event sponsor.

The SFL Robot Riot is an open Insect class fighting robot event conducted in conjunction with the STEM Tech Olympiad 2012 at the Miami Convention Center. Jim supports many fighting robot events by donating gift certificates from his store (the Robot Marketplace) and still competes in fighting robot events.

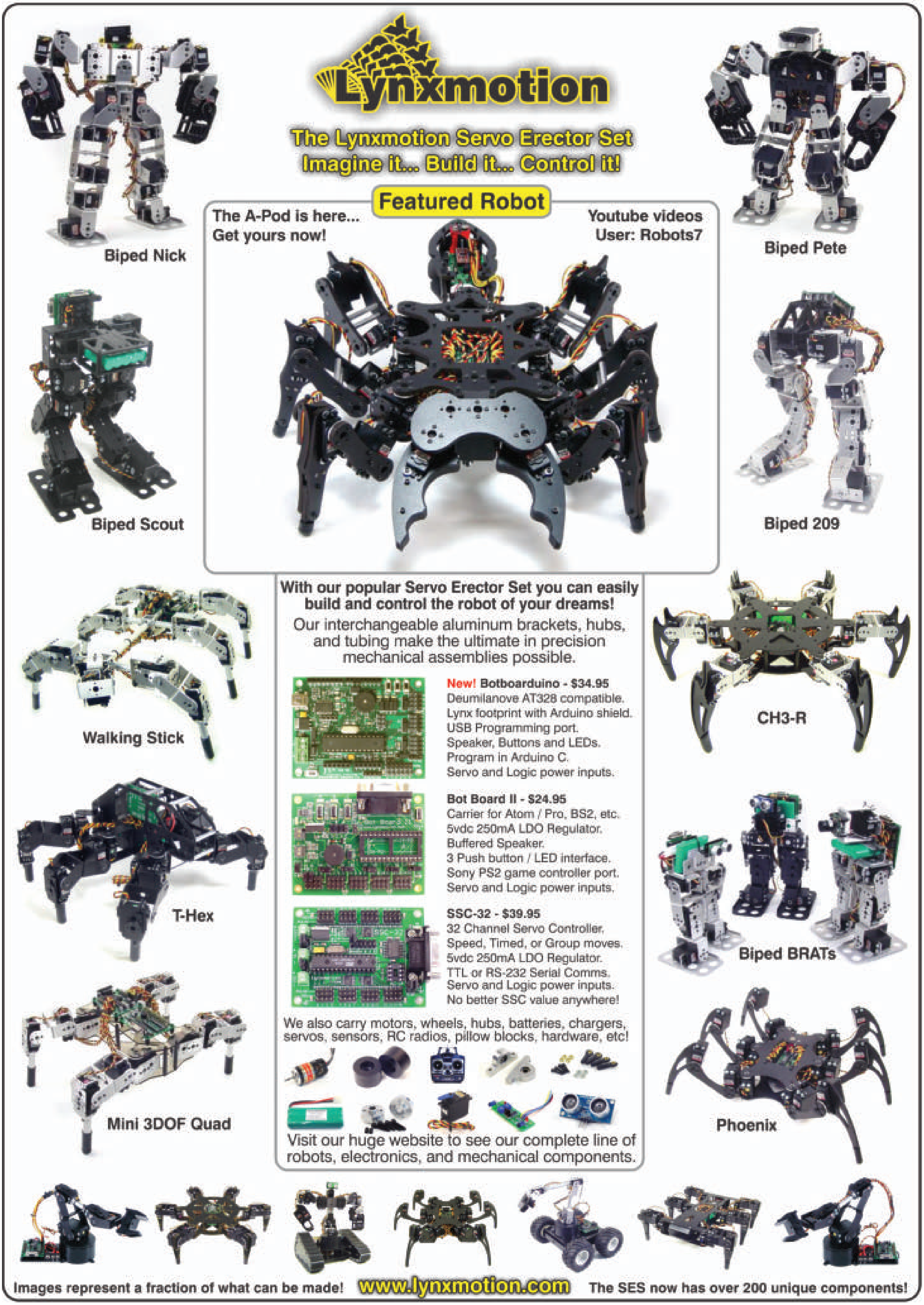
He had two bots entered in the SFL Robot Riot competition: a 150 gram Flea (a.k.a. Fairy) and a one pound Ant. Accompanying his dad was Jim's son Ian (**Figure 1**). We thought we were seeing the passing of the torch from father to son at the event.

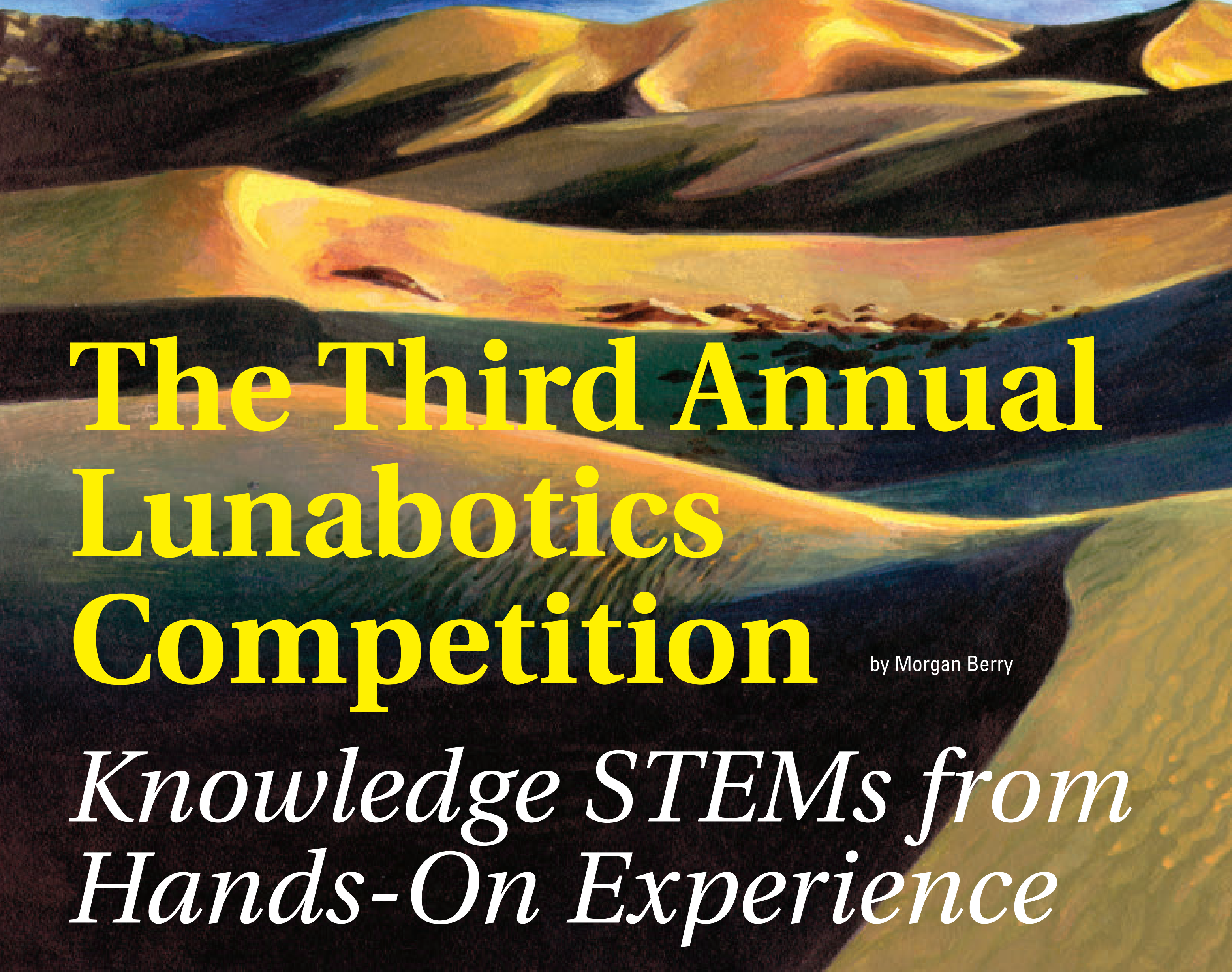
The torch actually turned out to be the candle on five year old Ian's birthday cake. The action was paused as those in attendance took time to sing Happy Birthday to him. I was expecting to see a cake decorated with the likes of Backlash or Nightmare, and quite frankly was disappointed to see Spiderman took top cake honors (**Figure 2**).

It didn't happen this time but I'm sure it won't be long until we see a second generation Smentowski at the controls of a fighting robot. And why not — he'll have the best pit man in the business. SV

FIGURE 1. Jim Smentowski with five year old son Ian.

FIGURE 2. Five year old Ian Smentowski with his Spiderman birthday cake.


Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Featured Robot
The A-Pod is here...
Get yours now!
Youtube videos
User: Robots7
Biped Nick
Biped Pete
Biped Scout
Biped 209
With our popular Servo Erector Set you can easily build and control the robot of your dreams!
Our interchangeable aluminum brackets, hubs, and tubing make the ultimate in precision mechanical assemblies possible.
Walking Stick
CH3-R
New! Botboarduino - $34.95
Deumilanove AT328 compatible.
Lynx footprint with Arduino shield.
USB Programming port.
Speaker, Buttons and LEDs.
Program in Arduino C.
Servo and Logic power inputs.
Bot Board II - $24.95
Carrier for Atom / Pro, BS2, etc.
5vdc 250mA LDO Regulator.
Buffered Speaker.
3 Push button / LED interface.
Sony PS2 game controller port.
Servo and Logic power inputs.
T-Hex
SSC-32 - $39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
Servo and Logic power inputs.
No better SSC value anywhere!
Biped BRATs
We also carry motors, wheels, hubs, batteries, chargers, servos, sensors, RC radios, pillow blocks, hardware, etc!
Mini 3DOF Quad
Phoenix
Visit our huge website to see our complete line of robots, electronics, and mechanical components.
Images represent a fraction of what can be made!
www.lynxmotion.com
The SES now has over 200 unique components!



The Third Annual Lunabotics Competition

by Morgan Berry

Knowledge STEMs from Hands-On Experience

Just this last May, a few hundred college students converged on Cape Canaveral — a town situated on the eastern coast of Florida. They spent the week forming memories with friends, making late night runs to the store, and frequenting the beach. Although this might seem like a typical college student spring break getaway, it was actually far from it. The students didn't go to Florida to party. They came to compete in the third annual Lunabotics competition hosted by NASA at the Kennedy Space Center Visitor's Complex.

The goal of the event is to encourage STEM (Science, Technology, Engineering, and Mathematics) education among college students, while also helping NASA develop an actual lunar rover prototype. The teams design and build a lunar rover that mines a simulated version of the regolith soil found on the moon. All those trips to the beach? They were doing last minute mining tests on their Lunabot. Those late night store runs? They weren't for beer. They were for replacement parts as teams prepared for the competition. These teens are some of the best and brightest in their academic fields, and NASA is taking full advantage of their talents.

Again, the primary goal of Lunabotics is STEM education. The teams learn engineering in a hands-on way by designing and building lunar rover prototypes. At the Lunabotics competition, they put their machines to the test in a pit of simulated regolith, competing with other teams to mine the most soil. In previous years, mining regolith was the only goal.

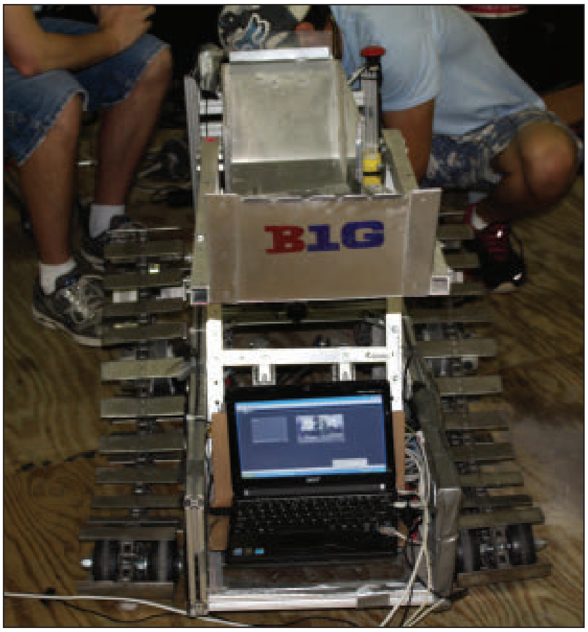
This year, NASA made changes to the competition. Now, in addition to the amount of regolith the Lunabots mine, teams also earn points for making a lighter and more compact bot, dust-proofing their machine, having a multidisciplinary team, and other criteria. For NASA, these changes mean more innovation in the team's entries. Every unique idea that the teams develop brings NASA one step closer to fine-tuning their actual lunar rover.

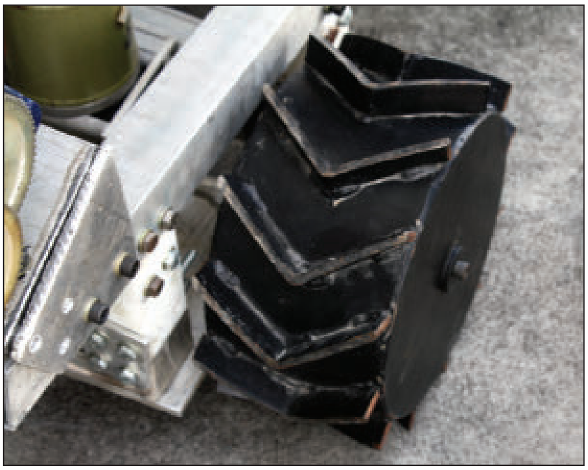
Students learn other skills for their future STEM careers in the competition, as well. The teams are required to write a systems engineering paper. This is basically a roadmap of how the students will design and build their bot. It is a frequent task of engineers in the real world; by learning this skill in Lunabotics, they will have a leg up as they enter into their chosen field. An outreach project — where teams are required to give presentations or demonstrations about Lunabotics in their local community — teaches the students how to educate others about their work. A slide presentation judged by NASA engineers is good practice for similar presentations that the students may do while working in their future careers.

So, what was it like to be a part of the 2012 Lunabotics competition? Check out this collection of photos from the event to find out.

Since NASA scientists won't have the luxury of being in the same room — or even the same planet — as a lunar rover, the same rules apply to the Lunabotics students. They must operate their bot from a command center bus, where they view the LunArena and their Lunabot on small, closed circuit television screens. The bots must operate wirelessly. One change that NASA made to the competition this year was giving the teams added points for making their bots autonomous — just as an actual lunar rover would be.

Team members from Florida State University (FSU) waited — excited and anxious — for their turn in the LunArena. The teams are required to mine a minimum of 10 kg of regolith, and pass through some lunar obstacles during their turn in the regolith pit. The FSU students decorated their suits with the school's logo, as well as with phrases such as "Go 'Noles!" (an abbreviation for the school's mascot, the Seminole), and "Hi mom!" Although wearing the thick white suits in the middle of Central Florida's 90° heat was uncomfortable — to say the least — they are a necessary safety requirement. The simulated regolith could easily get into the student's eyes and throats when they brought their robots into the LunArena.



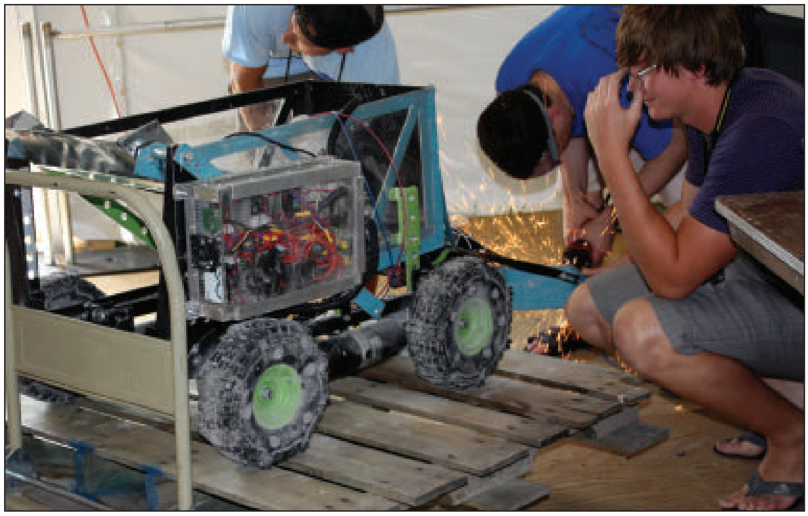


Last year, *SERVO* reported in an article entitled "Dust in the Wheels" (which, yes, was in fact riddled with some shamelessly bad classic rock puns) that one of the most common technical issues faced at last year's Lunabotics competition was traction problems. Many bots became trapped in deep ruts or slipped on top of the fine soil. The teams this year were more prepared for this problem. Every team *SERVO* spoke with had practiced and perfected navigating in the fine soil, making wheel problems a thing of the past.

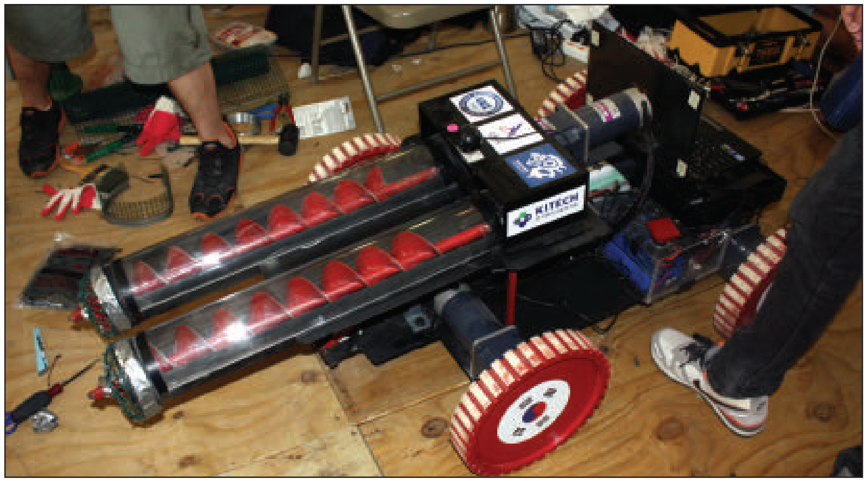
The pits for all 55 teams — a much needed air conditioned escape from the sweltering heat — were set up in a massive tent near the LunArena. Inside the pits were numerous large screens that showed the team's mining trips in real time. In addition to the chatter between teammates and the buzzing of electrical tools, frequent sounds heard in the pits were the gasps and cheers of members from other teams as they watched their competitors mine regolith. It appeared that the students enjoyed experiencing Lunabotics as a spectator sport as much as participating in the actual event.

After a particularly successful trip, one of the Colombian teams burst in through the entrance to the pits, chanting and dancing as they wheeled their Lunabot back in. At that moment, the atmosphere of the pits changed from a workshop into something more like the stands at an exciting sporting event. The change was tangible, and the other teams joined in the fun as well, dancing and clapping along with their competitors.

It wasn't all fun and games in the pits, though. As anyone who has ever attended a robotics event knows, some of the most frantic and difficult moments happen inside this small area. Teams make last minute adjustments, charge up their batteries, and mentally prepare for the competition. The difference between the teams who have already competed and the teams on deck is quite noticeable. Just a few stations over from the celebrating Colombian team was a team preparing for their impending round. They shifted nervously as they tested their robot just one last time before going off to face the LunArena.



The amount of time and energy the teams put into the competition is astounding. The teams spend months preparing for Lunabotics, and travel thousands of miles to compete. It takes a physical toll on the students. If you look closely at this picture of students from South Korea's Hanyang University, you can see one of the other members of the team asleep on the plywood floor in the background. No one said it would be easy, but for the students at Hanyang University competing at Lunabotics was well worth a little sleep deprivation.



The students from Hanyang University traveled about 7,500 miles across the Pacific Ocean and the Continental United States to make it to Cape Canaveral. They first learned about Lunabotics from a professor, who read about the competition on the Internet through his interest in NASA and space exploration. The team decided to participate in Lunabotics for their senior design project which — according to team member Min Yong Lee — they expect to earn a very good grade on. The team chose an auger design for their bot. They clearly focused on the bot's appearance as well as function, painting the auger, wheels, and other hardware a sleek red.

Buckets, and Augers, and Conveyors, Oh My!

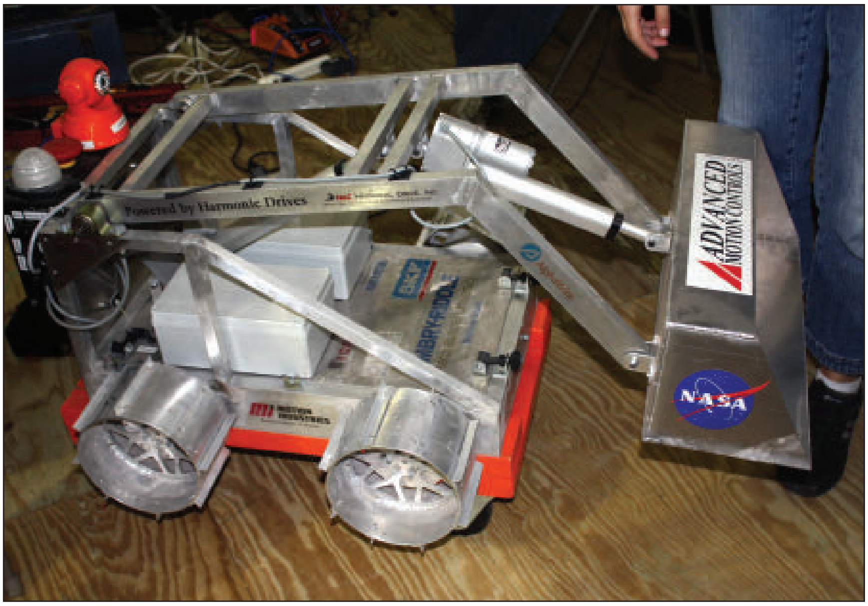
Three main robot designs have emerged out of the Lunabotics competitions. The bucket design works like a bull dozer, using a large scoop to collect the simulated regolith. The auger uses a screw design to pull the lunar soil up into a collection area. The conveyor transports lunar soil up onto a moving belt.

All three of these designs are inspired from real world applications. The advantages of the bucket are clear. This is a commonly used design for moving sediment on construction sites. It is a simple design which means less room for mechanical failures. However, this design requires constant trips to the deposit area.

By using an onboard storage area, conveyors and augers eliminate this problem. They also tend to look a lot more impressive than the simplistic bucket designs. As a result, however, they are more often plagued by technical problems than their scooping counterparts.

Augers have a tendency to become clogged with regolith. They do tend to stir up less dust than either the conveyor or the bucket design, though. This is important at Lunabotics because stirring up the fine lunar regolith can cause problems for bots, as well as cause point deductions.

As you can see, choosing a design style is a tricky business, with teams often starting and then scrapping designs as they wade through the many options. The winners of the mining competition have so far been conveyor designs but, of course, there's a lot more to winning than simply choosing a design. It takes a good design, skilled driving, practice, and a little luck to take home the first place prize.



Students from Embry Riddle Aeronautical University in nearby Daytona Beach used a bucket style design for their Lunabot. Their machine used Jaguar controllers and A123 batteries from DeWalt packs.

What set this bot apart from many of the other Lunabots was that all of the hardware for the bot was sealed inside the chassis using a gasket. This was in line with NASA's desire for the Lunabots to be designed in a dust-proof manner. Regolith is extremely fine and can easily cause problems for astronauts and lunar equipment alike; Apollo 17 astronaut Harrison Schmitt reportedly suffered from an allergic reaction after breathing in lunar regolith on the moon. The Apollo astronauts also had problems with the dust corroding and scratching equipment. The Lunabots designed with exposed parts would not last very long on an extended trip to the moon. By addressing this issue, Embry Riddle earned some extra design points.

Another positive design aspect for the team's Lunabot was its size. At 45 kg, this bot was about half the weight of some of the other competitor's entries.

The team from Iowa State University considered a number of different designs before deciding on a conveyor belt. They thought about using a bucket, but decided to challenge themselves with a more sophisticated design. They also made plans for a bucket-wheel excavator — where a machine collects regolith by scooping it with buckets attached to a spinning wheel — but calculated that they could mine more soil by using a conveyor design. Whatever the motive for choosing the design, it certainly paid off; the Iowa State team came in first place in the mining category.

Practice Makes Perfect

Since lunar regolith is so different from any soil on Earth, it can be difficult for teams to prepare for the unique qualities of the BP-1 regolith simulant. So, to practice for the event, teams have come up with some unique solutions to this difficulty.

Teams living in a coastal region have an automatic advantage over the landlocked teams. They have free, unlimited access to beach sand. Although sand is a lot different from regolith, it is cheap and abundant. It was very common for teams from Florida — as well as the coast of India and other countries — to practice in this manner. And, since the competition was held on the coast of Florida, many teams practiced at local Playalinda or Cocoa Beach the night before the event.

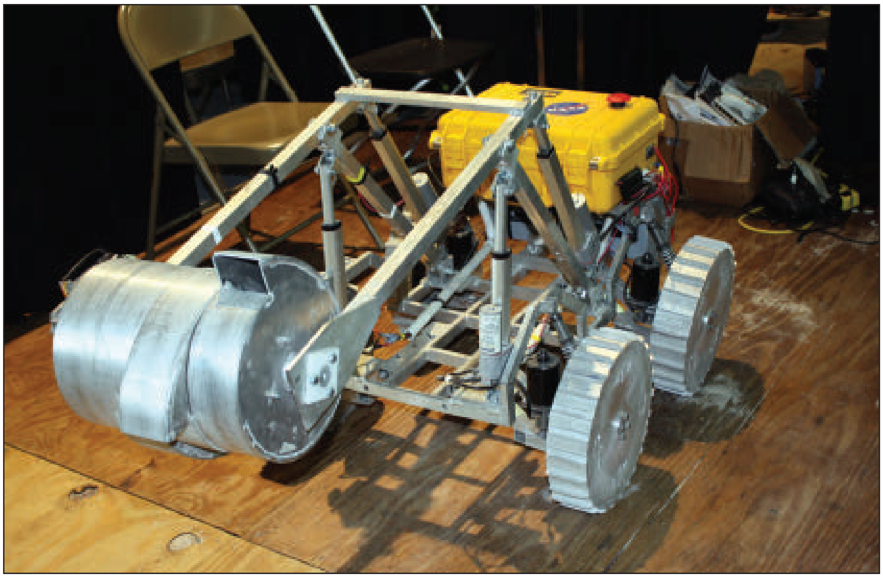
Another variation on this was the volleyball court method. Teams not near a beach often use the school's volleyball pit. For many schools, this is the only sand available to them. The Hanyang University team from Seoul, Korea practiced in a playground.

This is as far as most of the schools go to practice for the competition. Others go to some greater lengths to prepare.

Iowa State created their own mix to mimic the BP-1 regolith simulant. They used a mix of five parts Portland cement, three parts fly ash, and one part sand. They collected the fly ash from Iowa State's own coal burning power plant. Since the plant burns a different mix of materials every day, the team had to get all they needed on one day, so there wouldn't be inconsistencies in the material.

Last year, Laurentian University was sponsored by EVC Ltd., a company that produces simulant. They provided the team with regolith simulant to practice with. This year, Laurentian tested their bot with crusher dust — a fine dust that is the byproduct of some mining operations.

Not all teams stuck to one of the three most common design archetypes. This Lunabot from Montana State University chose a different approach. They designed a version of a bucket-wheel excavator, collecting the lunar regolith in a hollow drum. This kind of innovative design approach is exactly what NASA is trying to encourage with the Lunabotics competition. Montana State took home a first place prize for the systems engineering paper category and placed second in the outreach project category.



A major sponsor of the event this year was heavy machinery producer Caterpillar, Inc. This seems fitting, because Caterpillar products — like the pictured Cat® 287C semi-autonomous multi terrain loader — are basically designed with the same goals in mind as the Lunabots: to collect a large amount of soil. In fact, many of the Lunabotics teams modeled their bots after this type of machine, citing the effectiveness of Caterpillar machines in the real world at construction sites as evidence that the "scoop" design is effective.

"Caterpillar has a long history of supporting educational opportunities that promote the STEM areas. We need to encourage technology, innovation, and ingenuity to students of all ages. The development of autonomous systems will ultimately help our global customers boost safety, efficiency, and increase profitability," said Eric Reiners, a manager at Caterpillar, Inc., and one of this year's judges at Lunabotics. SV

The 2012 Lunabotics Winners

Joe Kosmo Award for Excellence (Grand Prize)
First Place — The University of Alabama in collaboration with Shelton State Community College
Second Place — Iowa State University in collaboration with Wartburg College
Third Place — West Virginia University

On-Site Mining Award
First Place — Iowa State University in collaboration with Wartburg College
Second Place — The University of Alabama in collaboration with Shelton State Community College
Third Place — Milwaukee School of Engineering

Judges Innovation Award
Polytechnic Institute of New York University

Efficient Use of Communications Power Award
Iowa State University in collaboration with Wartburg College

Best Use of Social Media
Universidad de Los Andes of Colombia

Slide Presentation and Demonstration Award
First Place — The University of Alabama in collaboration with Shelton State Community College
Second Place — West Virginia University
Third Place — Universidad de Los Andes of Colombia

Outreach Project Report Award
First Place — Iowa State University in collaboration with Wartburg College
Second Place — Montana State University — Bozeman
Third Place — John Brown University

Systems Engineering Paper Award
First Place — Montana State University — Bozeman
Second Place — John Brown University
Third Place — University of Illinois at Urbana — Champaign

Team Spirit Award
First Place — The University of Alabama in collaboration with Shelton State Community College
Second Place — Instituto de Astrobiologia Colombia IAC
Third Place — Polytechnic Institute of New York University

Meet the Parrot AR Drone 2.0

by Bryan Bergeron

www.servomagazine.com/index.php?/magazine/article/august2012_Bergeron

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

FIGURE 1. X-ray view of the AR Drone with its cloverleaf indoor hull.

Drones — especially the quadcopter variety — are hot. Not only are they cool to watch and to fly, but they're excellent R&D platforms for robotics. In this article, I'll examine the Parrot AR Drone 2.0 ('AR Drone'): an entry-level quadcopter that can give you an introduction to the many advanced technologies under development for the larger quadcopter platforms. Figure 1 shows the lower fuselage of the AR Drone with the expanded polypropylene indoor hull overlaid.

Quadcopters

If you've been to one of the Maker or SparkFun fairs recently, you know that quadcopters rule when it comes to autonomous vehicle competitions. One reason for the interest in quadcopters is that — when properly designed — they are the epitome of mechanical elegance. Instead of the complicated gyros and gearboxes of a helicopter or the ailerons and other control surfaces of an airplane, there are four fixed electric motors and props. Change the relative speeds of the props and you control the flight of the craft.

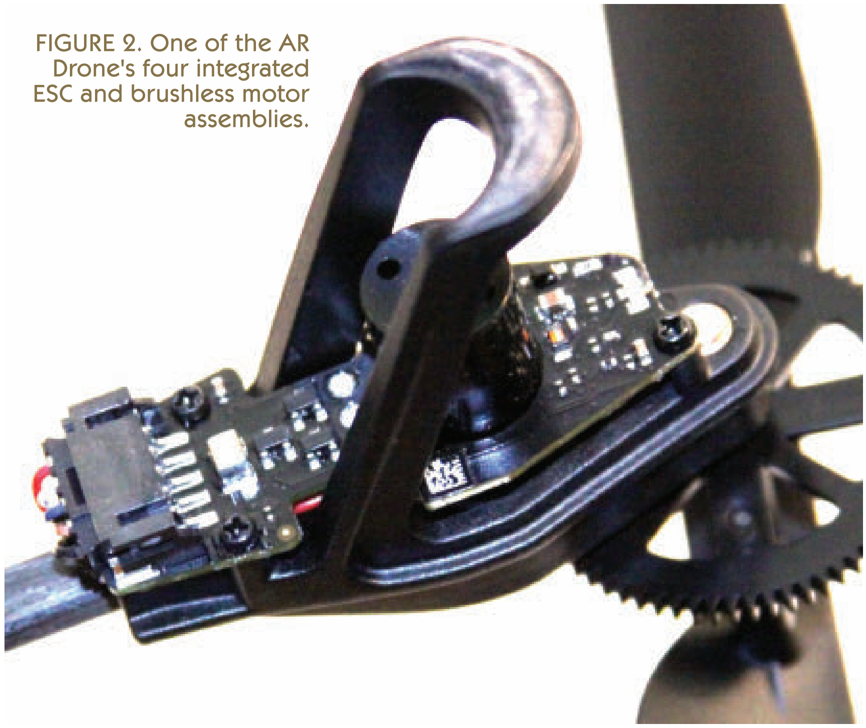
This elegant, efficient platform is possible because the simple hardware is paired with powerful software, and because of the consistent availability of high speed brushless motors and electronic speed controllers (ESCs). **Figure 2** shows one of four integrated ESC/brushless motor assemblies on the AR Drone.

You don't need a degree in mechanical engineering and access to a machine shop to design and build a quadcopter. You do need to understand sensors, their limitations, and some advanced sensor handling algorithms if you're going to design your own. **Figure 3** shows the underbelly of the AR Drone. From the top to bottom of the image are the low-res camera, ultrasonic range finder, and 720p camera.

Of course, I'm using a four-motor quadcopter as a generic term for electronically controlled, direct drive flying platforms. There are copters with three or six motors arranged in a 'Y' configuration; with four or eight motors arranged in an 'X' configuration; with six or 12 motors arranged in a "✭" configuration; and so on. When a configuration has double the motor count, motors are paired vertically with one pushing and one pulling.

If you're itching to work with a quadcopter, then consider the AR Drone as a gentle introduction to the platform. The quadcopter is relatively inexpensive — I paid $299 at Amazon. However, the price doesn't include a controller – a compatible IOS or Android device. I use an iPad.

FIGURE 2. One of the AR Drone's four integrated ESC and brushless motor assemblies.



Video Games

The AR (Augmented Reality) Drone is first and foremost a video game peripheral. It's not intended to serve as a robotics platform, and Parrot doesn't support hacking their copter. It's a closed software platform, unless you happen to be a game developer.

Many design features reflect concern for the safety of young video game players versus performance or ease of modification for robotics enthusiasts. There's the ample expanded foam padding to protect the electronics from the environment and to protect pets and humans from flying electronics. There's a 1,000 mAh lithium-ion battery encased in heavy, hard plastic (see **Figure 4**), and a geared, relatively inefficient propulsion system. The advantage of this focus on safety — besides international access to toy stores — is that you can use the AR Drone just about anywhere without concern for inflicting (much) harm on bystanders. You certainly can't say that about the larger quadcopter platforms.

Hardware

The hardware inventory of the

Brushless Motors

Brushless motors are essentially high speed, high efficiency stepper motors. Compared with traditional brushed DC motors, brushless motors produce higher torque by weight, operate at higher efficiency, produce less acoustic and electrical noise, and have a greater mean time before failure. Of particular importance to flying vehicles is torque/weight and efficiency which relate directly to thrust and battery life. Brushless motors are rated in kV, the rpm constant, or number of revolutions per minute that the motor will turn when 1V is applied to the motor with no load.

FIGURE 3. Underside of the AR Drone showing the 720p camera, ultrasonic sensor, and low res camera.

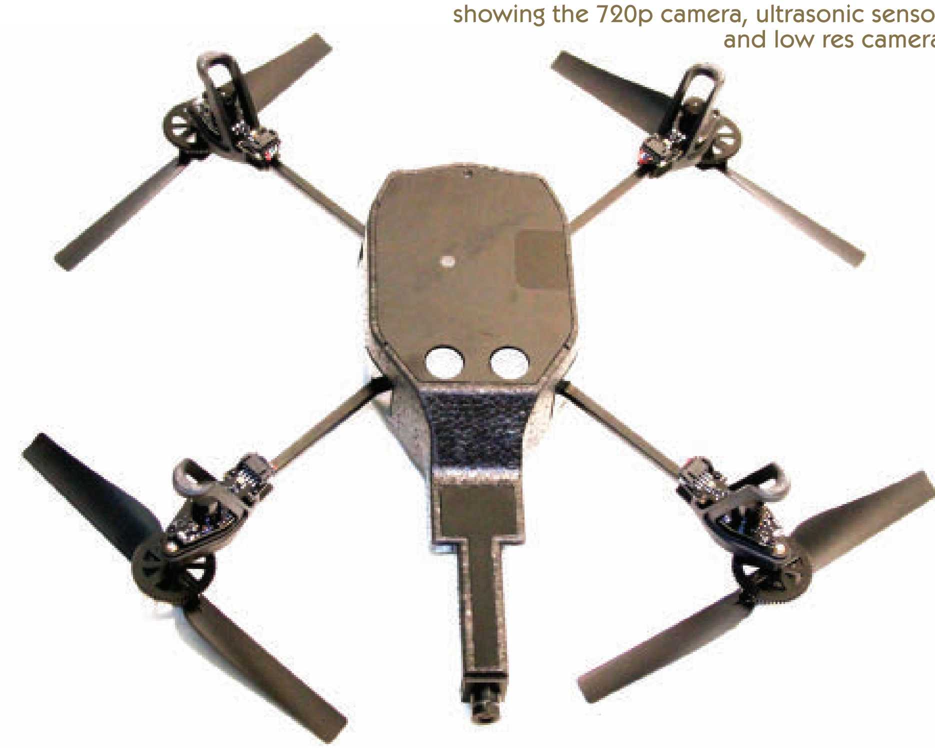


FIGURE 4. Stock lithium-ion battery mounted in main fuselage. The USB connector is for storing video files.

AR Drone reads just like that of a smartphone: 1 GHz, 32-bit ARM processor, 1 GB RAM, 720p 30 fps HD camera with video streaming, and Wi-Fi, all running under embedded Linux. The onboard sensor complement includes an ultrasonic range finder, an atmospheric pressure sensor and ultrasonic range finder to determine altitude, a three-axis gyroscope, magnetometer, and three-axis accelerometer. In addition to the forward-facing 720p video and still camera, there is the equivalent of an optical mouse – a downward facing, low resolution camera for tracking ground speed and for hovering motionless.

The large indoor hull protects the props from hitting walls, curtains, and fingers. The outdoor hull is an expanded foam top that fits on the fuselage. Either hull essentially creates a lightweight, flying marshmallow that is best suited for indoors or outdoors on a dead calm day.

The electronic speed controllers are matched and tuned to the four brushless motors, pulsing at 200-300 Hz. Whereas most quadcopters on the market use direct drive, the AR Drone uses a step-down gear system to drive the specially-designed slow speed props. As a result, I can stop the AR Drone dead in flight by brushing a prop with the back of my hand – all without injury.

Touch one of the AR Drone blades and you'll feel a modest bite before it immediately shuts down and falls to the ground. I've tried this several times – it's not pleasant, but it doesn't leave a mark. Of course, a prop to the eye can cause blindness. Perhaps that's why the AR Drone is rated for children 14+.

Teardown

The 2.0 version of the AR Drone is less teardown friendly than the original. The bottom plate — which is made of thin plastic — is glued to the expanded foam with a super-tacky adhesive that takes some of the foam with it. The same goes for the stamped aluminum supports on the sides and top of the fuselage (see **Figure 5**). If you have strong fingernails, you can remove the supports, but don't expect a clean extraction. In other words, once you take it apart, you'll never get it back to the original state. Getting to the 720P camera requires that you peel off the top aluminum plate as in **Figure 5**.

Once inside, you'll find the circuit board with sensors and CPU are tucked away in a cozy bed of expanded polypropylene (see **Figure 6**). The board is held in place by four screws attached to the plastic battery compartment. Remove the four plastic adhesive patches from where the beams enter the expanded polypropylene hull and peel open the sides to extract the board and crossbeam assembly.

FIGURE 5. Extracting the 720p forward-facing camera requires removal of the top aluminum plate. Note the foam pieces adhering to the removed plate.

Figure 7 shows the circuit board still attached to the battery holder on the opposite side of the crossbeam. Also visible in the **photo** are the 720p camera (far right), ultrasonic range finder (lower middle), and the low resolution, downward facing camera (top center).

With the foam out of the way, removing the four retaining screws releases the plastic battery holder, allowing you to separate the board and sensors from the bare frame as shown in **Figure 8**. If you look carefully, you can see the small nib on the left side of the center support. The nib faces forward, and the connectors are toward the rear of the craft.

That's all there is to the teardown. If you have strong nails, it shouldn't take more than 5 minutes, tops.

As with other aircraft, weight matters. Every gram of excess weight requires more battery power or eats into available flight time. Here are my weigh-in results:

- Body (intact) 262g
- Battery 103g
- Fuselage (bottom only) 31g
- Electronics 50g
- Bare frame with motors and props 187g

So, the contributors of weight to the intact AR Drone — in order of decreasing weight — are the bare frame with motors and props, battery, electronics, and fuselage bottom.

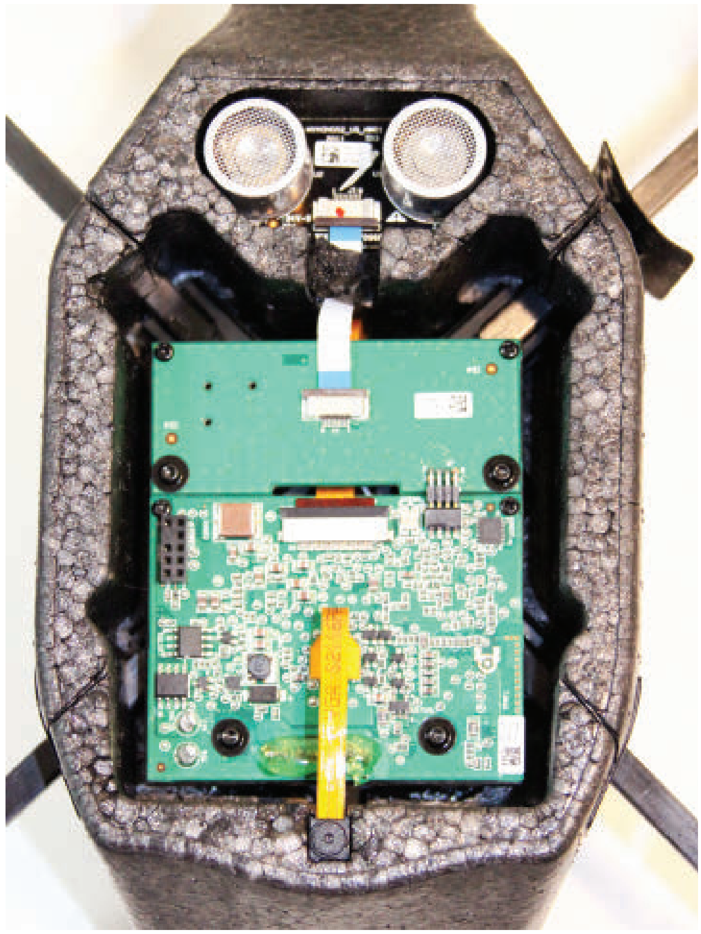
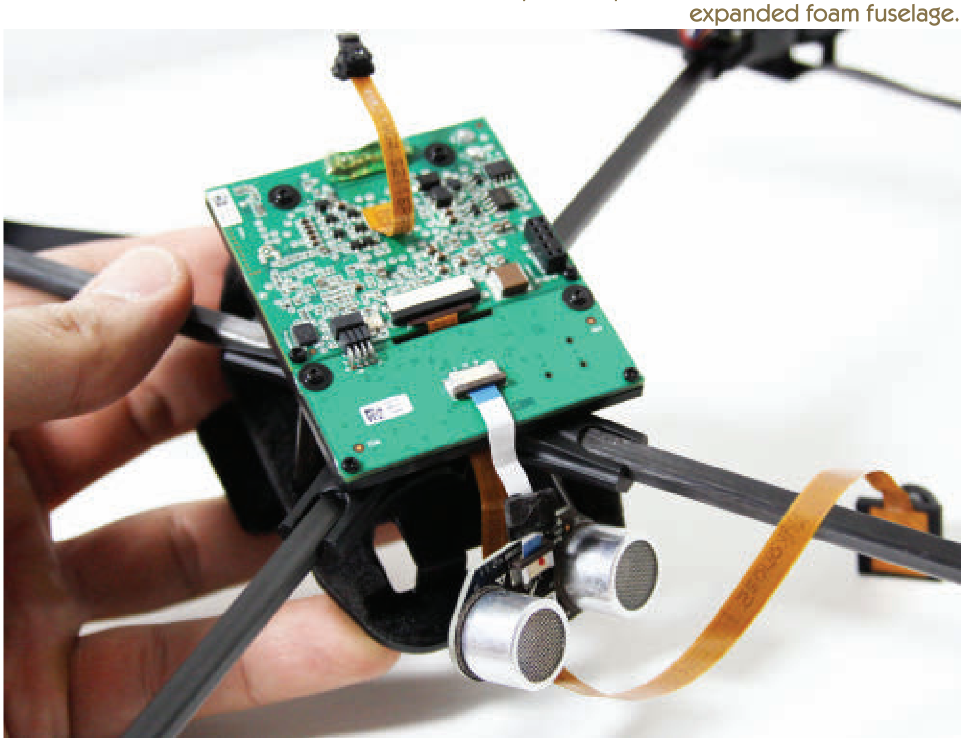


FIGURE 6. AR Drone with underside plastic panel removed. Note the programming plug on the left and the downward-facing camera at the bottom of the figure.

Assessment

Thanks to the AR Drone firmware and the software on my iPad, operation was a breeze. I was up and running in five minutes after ripping open the carton. It helped that I had run through the simulator, which is available as a free download on

FIGURE 7. Circuit board, sensors, and frame after removal of the expanded foam fuselage.



Electronic Speed Controllers

Electronic speed controllers (ESCs) convert DC voltage — say, 11.1V — to a three-phase electric current. The PWM duty cycle of the current — which is determined by a servo signal from a microcontroller — determines the speed of the attached brushless motor. At 100% duty cycle, the ESC output is essentially the input voltage, or 11.1V in this example. At 50% duty cycle, the ESC output is 5.55V, and the speed of the brushless motor is half of maximum.

FIGURE 8. Bare 'X' frame of the AR Drone. The craft is facing left, as indicated by the small nib just visible on the supporting brace.

iTunes. The simulator (shown in **Figure 9**) is a good approximation of the real thing, but is far from perfect.

Using the controller program, touch the 'take off' button and the AR Drone takes to the air and hovers in place about three feet off the ground. Next, place your thumb in the left circle and tilt your smartphone or tablet in the direction you want the AR Drone to go. Increase or decrease altitude or rotate the drone with your right thumb. If all else fails, simply remove your thumbs from the screen. The AR Drone will stop — dead in its tracks — and hover. The other option is to hit the 'emergency' button which causes the craft to dive towards the ground.

For the price, the functionality is amazing. Auto-land, auto-takeoff, and stationary hover are lifesavers. The Wi-Fi control is crisp and the video lag is imperceptible. My only reservations are regarding the noise and battery life.

This isn't something you can use to sneak up on a squirrel or bird for a close-up. (Think Neato or Roomba vacuum cleaner.) The gears simply grind away. Battery life is a mere seven minutes. Moreover, the lithium-ion battery requires 90 minutes to fully charge on the stock dumb charger. Leave the battery in for longer and risk explosion. Extra batteries are $40. There isn't much you can do about the sound level, but the battery limitations are easily remedied.

Battery Mods

Assuming you fall in love with the AR Drone, your first mod or at least next purchase should be more and different batteries. The stock battery and charger are atrocious. The battery is low density, has low current capacity, and the charger will kill it if you don't unplug the charger after 90 minutes.

You could spend $40 for additional stock batteries, but that would be a waste. I bought four 1,300 mAh/11.1V 40C Blue LiPo lithium-ion batteries from **XHeli.com** for less than $50, including shipping. The Blue LiPo battery has the same dimensions as the stock battery (see **Figure 10**) and is a few grams lighter. There's nothing special about this brand; just make certain the voltage and C rating are adequate, and that the size and weight are approximately the same as the stock battery.

The standard battery is rated at 1,000 mAh/11.1V and only 10C continuous discharge. The C rate is the rate at which a battery is discharged relative to its maximum capacity. A 1C rate is equivalent to discharging the entire battery capacity in one hour; 10C means that the stock battery can discharge at a rate of 10 x 1,000 mA or 10A. In contrast, the Blue LiPo's continuous discharge rating is 40 x 1,300 mA, or 52A.

FIGURE 9. AR Drone simulator interface. After takeoff, place your thumbs on the black buttons and start flying.

The Blue LiPo line of lithium-ion batteries comes with 4 mm banana connectors. Other brands may use these or different connectors, but that's not a problem. You can either buy an adapter or simply swap out the original connector with 4 mm bullet connectors ($2 for a pack of six; **www.HobbyKing.com**). Just note the polarity difference for male and female on the battery and AR Drone connector. Also protect the battery connectors with an insulating covering or you'll end up with a shorted lithium cell.

You'll also need a smart lithium-ion battery charger. Almost anything will be better than the stock charger – expect to spend at least $35-$50 for a good one. You can still keep your original battery and charger as a spare. Just update the battery's output connector with 4 mm bullet connectors.

Less Friction

Your second consideration for a mod should be to replace the brass bushings supporting the prop shaft with real bearings (see **Figure 11**). There's a significant aftermarket for bearings for the AR Drone, and for good reason. After two hours of flight time, the lower side of the bushing on my quadcopter had significant signs of wear from the stainless steel prop shaft. A set of eight stainless steel bearings will cost you about $20 on eBay.

When you change the bushings for bearings, make the change one engine at a time. Don't confuse which prop goes where because two engines turn clockwise and two turn counterclockwise to balance out the craft.

Less Wind Drag

The AR Drone is akin to a flying marshmallow, or perhaps a flying Twinkie. The expanded foam hull is great for absorbing impact and for keeping spinning props from little fingers. However, if you want to fly your AR Drone outside in a modest breeze and add some zip into your indoor maneuvers, then lose the bulky foam.

After the teardown, I decided to discard the foam hull and go for minimal wind resistance. I started by weighing various plastics and other building materials. I settled on a transparent plastic CD-ROM case for the bottom and a 3" diameter 1/16th thick birch disc for the top (see **Figure 12**).

FIGURE 10. Stock battery vs. the smaller, higher capacity Blue LiPo battery. Gold 4 mm bullet connectors can be used to modify the stock battery.

FIGURE 11. Original brass bushings and stainless steel replacement bearings. The special tool in the upper left is used to remove the props.

FIGURE 12. Building materials for the low cross-section fuselage. The original plastic bottom is used as a template for the CD-ROM case.

FIGURE 13. AR Drone in flight with DIY low-drag fuselage.

To create the bottom support, trace the circuit board outline onto the CD case with a marker and then cut out the pattern. Drill matching holes for the top disc, and mount the Velcro® battery strap with a few stitches of thick thread run through the disc. Place a small square of foam between the circuit board and crossbeam, and between the disc holding the battery and the 'X' brace; compress the sandwich with tie wraps.

Figure 13 shows the new and improved AR Drone in flight. I took the photo unaided by placing the AR Drone in hover mode. The drone hovered there until it nearly exhausted the battery and automatically landed.

Position the battery so that the center of gravity is right above the center of the crossbeam. Then, attach a 1/2 inch Velcro dot on the battery's body so that it lines up with the Velcro strap. Attach additional dots to each battery in your collection. The dots keep the battery from sloshing around during flight or during a crash. The lean and mean AR Drone is zippy and maneuverable, even with a modest breeze. If you're into the stealth look, paint the top disc black and use a darker CD-ROM case. I used white tie wraps and an unpainted disc so that you could see the construction details.

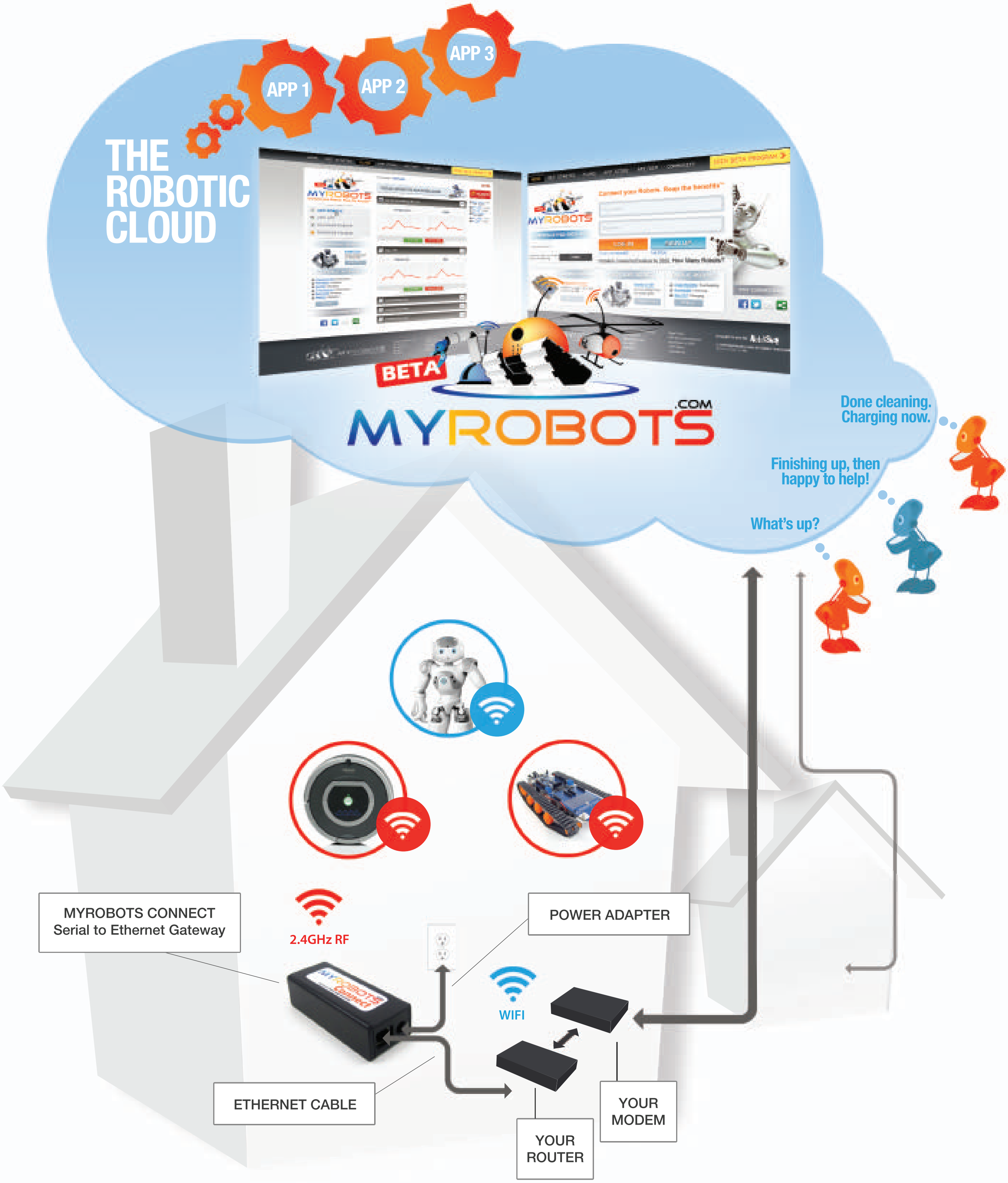
Greater Range

You can abandon the phone/tablet interface and 100 meter range and go with a standard R/C unit with a range of 1+ Km. Technically, the mod is relatively easy – check Google and YouTube for detailed instructions – but if you take your quadcopter up a Km or more, then you're entering 'serious' R/C territory with potential legal pitfalls. Avoid this mod if you plan to fly your quadcopter in a city or near an airport.

Bottom Line

For the price of a basic R/C transmitter, you get a full featured quadcopter that's a great starter platform. It's quite a bargain, especially if you already own a compatible iOS or Android device. At a minimum, consider upgrading the battery system to a smart charger and less expensive, higher capacity lithium-ion batteries from an online R/C shop.

There's no need to feel left out of the robotics R&D community if you live in a big city or near an airport, or simply can't afford to move up to a larger quadcopter platform. There's a large and growing community of enthusiasts bent on transforming both the AR Drone and AR Drone 2.0 from toy to workhorse. Take a look at **dronehacks.com** for a sample of the work underway on this platform – from voice control with an iPhone to full-body steering with a Kinect. And don't miss (from *Wired*) Chris Anderson's excellent DIY Drones podcast series. **SV**



Connect your robots. Reap the benefits ™

MyRobots.com strives to make cloud robotics a reality accessible to everyone and everything by enabling all robots and smart devices to connect to the Internet.

You can think of MyRobots.com as a social network for robots and smart devices. In the same way humans benefit from socializing, collaborating and sharing, robots can benefit from interacting and sharing their sensors' information, which provides insight on their current state, and allows them to be controlled and monitored remotely.

Through the **MyRobots App Store**, robots augment their capabilities so they can transcend their limitations, allowing them to do more than what they were originally designed for. This means robot owners can reap the full benefits of their robots.

Reach Out and Touch Something

Part 2 – Integrating Touch Sensing With Your Robot

by Gordon McComb

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com

Touch is the most basic of all senses — for humans and robots. Better yet, adding touch to a robot is easy on the pocket book; touch sensors are among the least costly parts you'll have to buy for your bot. For under $15, you can equip your wandering beast with a variety of touch detection devices.

Last month, we looked at three common and inexpensive touch sensors that can be added to your robot. We discovered what they were and how they connected to a microcontroller such as the Arduino. In this second and last installment, you'll learn how to integrate these sensors into a motorized robot base. Each type of sensor detects contact with a physical object. Knowing it has collided with something, the robot will immediately back off and try another route.

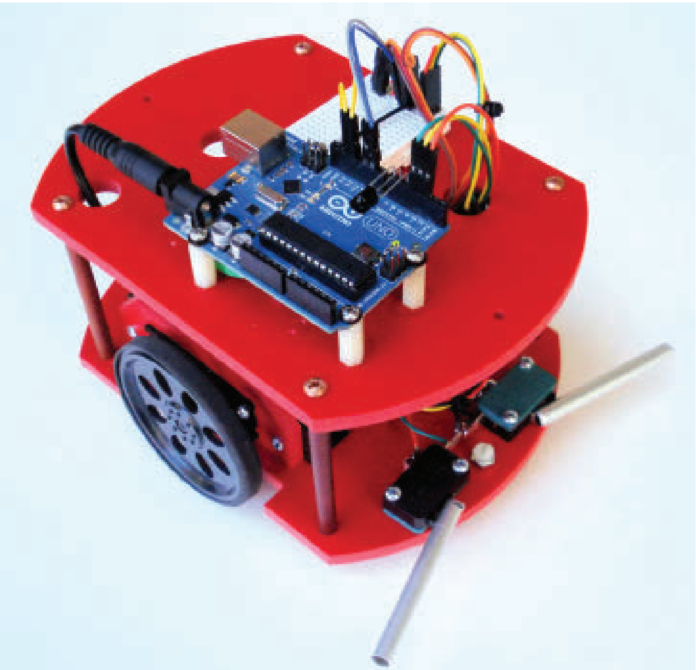


FIGURE 1. ArdBot expandable robot with leaf switches attached. The easy-to-drill plastic pieces of the base allow for easy experimentation.

A Basic Base for Basic Experimenting

It's always best to start from a known. For a demonstration robot, I'll use an ArdBot chassis from my online store Budget Robotics (see the **Sources** box). The ArdBot is an expandable two-wheeled differentially-steered robot that uses the popular Arduino Uno (or compatible microcontroller) as its main brain. The kit comes with body pieces and assembly hardware only — you add your own Arduino, batteries, continuous rotation servos, and wheels.

The ArdBot comes in two versions. The standard ArdBot has a larger 7" round bottom; the ArdBot II has a streamlined bottom deck. **Figure 1** shows the ArdBot II. On both versions, the motors and batteries go on the bottom deck.

To simplify the power requirements, the ArdBot uses separate battery supplies for the Arduino and the servos. The Arduino is powered by a nine volt battery connected to the power jack on the board, while the servos are powered by their own 4 x AA pack. You can use either non-rechargeable or rechargeable AA cells. If you use rechargeable batteries, remember that they put out a slightly lower voltage (1.2 volts as opposed to 1.5 volts), so the servos will run a little slower.

There's room on the top deck of the ArdBot for the Arduino, a small mini solderless breadboard, and assorted

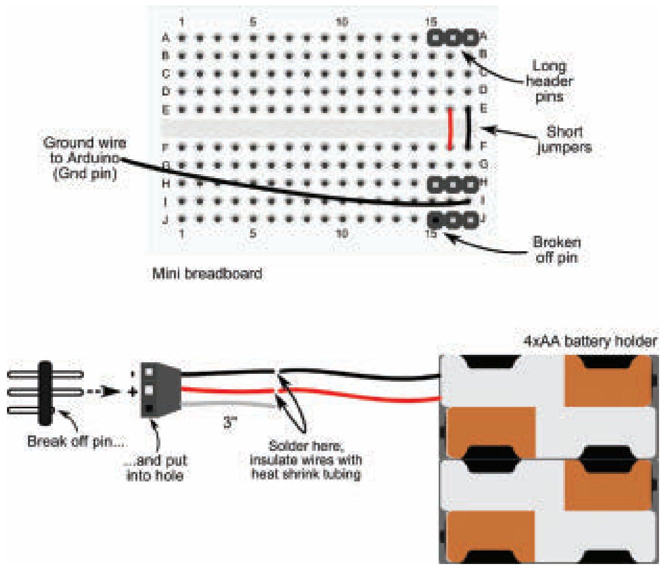


FIGURE 2. Solderless breadboard wiring diagram for connecting the servos and power on the ArdBot.

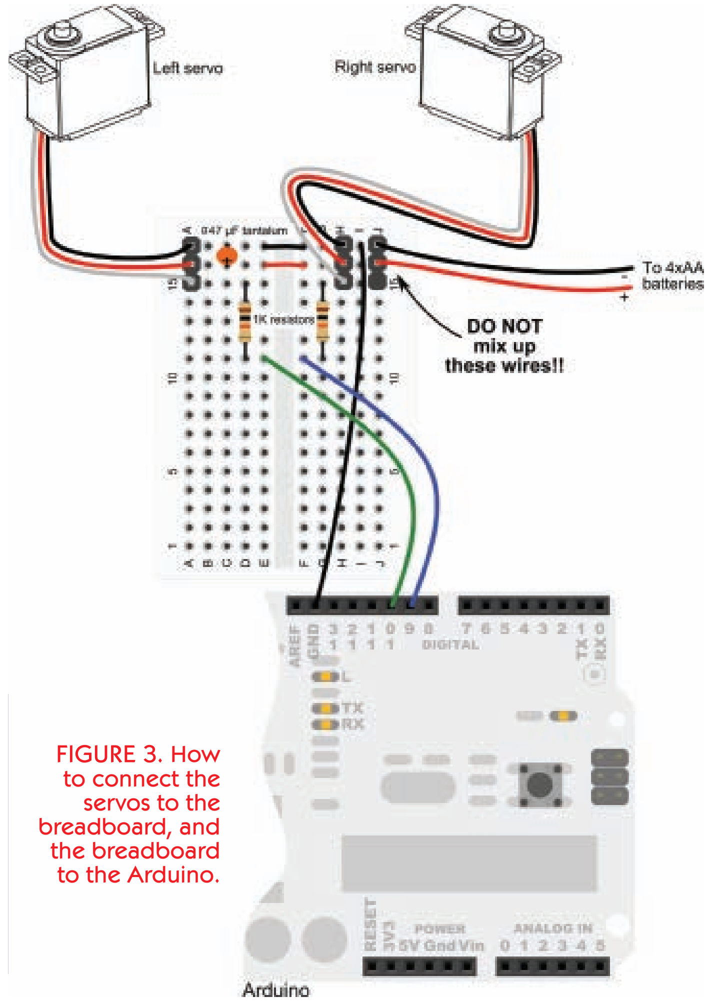


FIGURE 3. How to connect the servos to the breadboard, and the breadboard to the Arduino.

other electronics. The two decks are separated by standoffs. Of course, you can also use most any other desktop robot for this project, like the Parallax Boe-Bot. The main requirement is that your robot sports a pair of servo motors for scooting around the floor. Obviously, you'll need to adjust the construction details accordingly.

Figure 2 shows the basic breadboard wiring for the ArdBot. Three double-long male header pins act as connectors for the servos and servo battery pack. Jumpers connect the header pins to one another, and to the Arduino. For the battery connection, clip off 3" from the end of a 12" three-wire servo extension cable. Set the 9" length to one side — you'll use it later — and solder the 3" length onto the pigtail ends of the AA battery holder. Use heat shrink tubing or insulating tape on all solder joints. (This is all assuming the battery holder has 6" wires. If this is not the case, you'll need to use a longer length of servo extension cable.)

Figure 3 depicts how to wire the servos to the breadboard, and then connect the Arduino. Note the extra components on the breadboard:

- The 47 µF tantalum capacitor aids in removing voltage spikes caused by the servos starting and stopping.
- The twin 1 KΩ resistors help resolve an issue with the Arduino related to using servos and sensors with a high impedance (more about this in a bit).

Bumper Car Bots With Leaf Switches

Leaf switches are like cat whiskers: Brush against the leaf, and the switch contact closes. That's all your robot needs to know when it's made contact. Two standard leaf switches mounted to the front of your ArdBot let it detect when it's bumped against something. With the switches situated to the sides, your bot can determine if the object is on the left or on the right, and so can steer around it.

Figure 4 shows a pair of leaf switches mounted like bumpers to the front of the ArdBot. You can find switches like these at many online electronics outlets. They're also

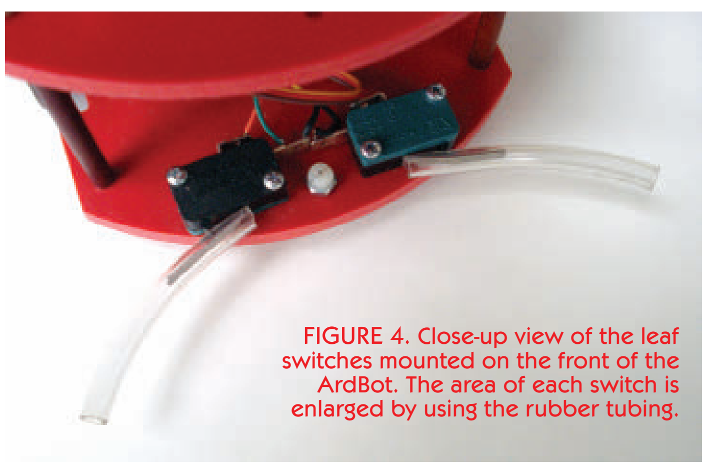


FIGURE 4. Close-up view of the leaf switches mounted on the front of the ArdBot. The area of each switch is enlarged by using the rubber tubing.

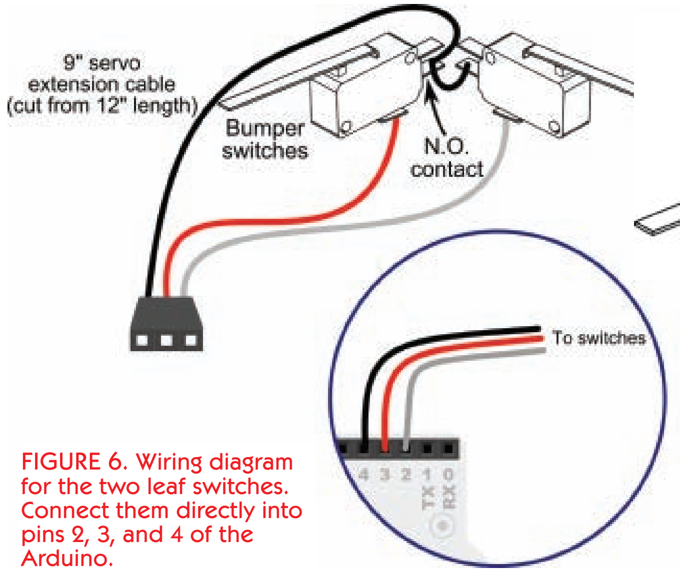


FIGURE 6. Wiring diagram for the two leaf switches. Connect them directly into pins 2, 3, and 4 of the Arduino.

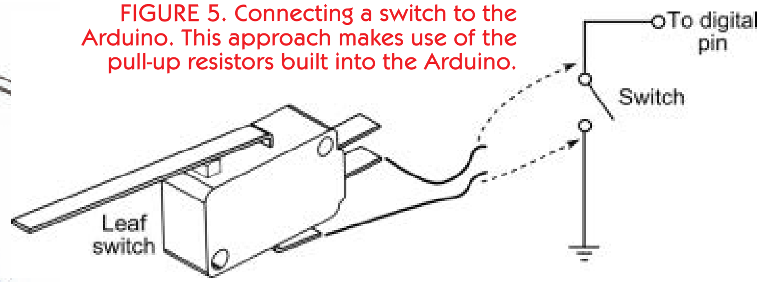


FIGURE 5. Connecting a switch to the Arduino. This approach makes use of the pull-up resistors built into the Arduino.

common as surplus. The ones in the picture are available from All Electronics (a *SERVO Magazine* advertiser) for about $2 each. The length of the leaf varies from one switch to another. If you'd like a larger contact area, you can always augment it by adding a 2" to 4" aluminum or rubber tube over the leaf (like that in the **figure**). Cut the tubing to length, then slip it over the leaf. You'll need to gently bend the leaf outward to make room for the added thickness of the tubing. Make sure that the button on the switch is open when there is no force against the leaf.

Mount the switches by finding two suitable holes in the base of your robot (these are already provided on the ArdBot) or drill new ones. Most leaf switches have three connections: common, normally open (NO), and normally closed (NC). Wire the common and NO connections. If space is tight, break off the NC connection to make room.

Refer to **Figure 5** for a schematic diagram for adding switches to the Arduino. Notice that this is slightly different than the circuit in Part 1 of this article. There's no pull-down resistor. This variation takes advantage of the pull-up resistors already built into the Arduino. As you'll read later, under software control you can activate the pull-ups, saving you from having to use external resistors.

Make a wiring hardness for the switches by using the remaining 9" length of the three-wire servo extension cable you cut earlier. Solder the wires of the cable to the switches as shown in **Figure 6**. Use a set of double-long male header pins to attach the other end of the servo extension to pins 2, 3, and 4 of the Arduino. Digital pins 2 and 3 serve as inputs from the switches; pin 4 provides a common ground connection for the switches. Code running on the Arduino makes pins 2 and 3 inputs, and pin 4 an output. Pin 4 is then set LOW — equivalent to ground.

Listing 1 shows a demo sketch that helps you test the servos of the robot to be sure the basic wiring is correct. To use, place the robot on a book to lift its wheels off the ground. Upload the sketch to the Arduino, then plug in the 4 x AA battery pack for the servos (be absolutely sure not to cross the servo wiring!). Leave the USB cable between your PC and Arduino connected. Press the reset button on the Arduino, and watch the wheels turn. The

LISTING 1 — Servos_only.ino.

```
#include <Servo.h>

Servo servoLeft;              // Define left servo
Servo servoRight;             // Define right servo

void setup() {
  servoLeft.attach(10);       // Set left servo to pin 10
  servoRight.attach(9);       // Set right servo to pin 9
}

void loop() {
  forward();
  delay(2000);
  reverse();
  delay(2000);
  turnRight ();
  delay(2000);
  turnLeft();
  delay(2000);
}

// Motion routines for forward, reverse, turns, and stop
void forward() {
  servoLeft.write(0);
  servoRight.write(180);
}
void reverse() {
  servoLeft.write(180);
  servoRight.write(0);
}
void turnRight() {
  servoLeft.write(180);
  servoRight.write(180);
}
void turnLeft() {
  servoLeft.write(0);
  servoRight.write(0);
}
void stopRobot() {
  servoLeft.write(90);
  servoRight.write(90);
}
```

servos should rotate clockwise and counterclockwise as the sketch goes through its paces.

If things are working so far, disconnect the USB cable and plug in the 9V battery to the Arduino. Press and hold the reset button on the Arduino, and place the robot on the ground. Release the button, and the robot should go into its servo demonstration mode.

Note that "front" and "back" (and "left" and "right") are somewhat objective in a robot like the ArdBot. Either end can be the front, so left and right is relative. If your robot seems to behave opposite to what it should — that could happen if the wiring gets crossed, or you use servos that rotate in the opposite direction than the "standard" — swap the values in the motion routines (forward, reverse, etc.).

Now you're ready to test the bumper switches. Reconnect the USB cable, upload **Listing 2**. Disconnect the USB. Place the robot on the ground, and momentarily depress the reset button to restart the sketch. The bot should move forward. Test the switches by touching one or the other.

- When the right switch is momentarily depressed, the robot should back up briefly, turn to the left, then continue forward again.
- When the left button is momentarily depressed, the robot should likewise backup, but turn to the right before moving forward.

The sketch in **Listing 2** uses several unique features. First is how the switches are directly connected to the Arduino. This is accomplished using the built-in pull-up resistors of the Arduino which obviates the need for external resistors. Further, the ground connection for the switches is derived from pin 4 which is set as a LOW (zero volts) output.

Second is that the sketch uses interrupts to note whenever a switch has been struck. The interrupts are set up with the two lines:

```
attachInterrupt(0, hitRight, FALLING);
attachInterrupt(1, hitLeft, FALLING);
```

which tell the Arduino to trigger an interrupt event whenever the signal on pins 2 or 3 falls from HIGH to LOW. Note that on the Arduino, interrupts are referred to by number, not by pin. Interrupt 0 is internally attached to pin 2; interrupt 1 is attached to pin 3.

Each interrupt must have a corresponding interrupt handler. This is by the hitLeft and hitRight functions at the end of the sketch. These functions are automatically called whenever an interrupt occurs. Both functions merely set a flag variable (pbLeft and pbRight) whenever the code passes through the interrupt. At each loop through the main program, the Arduino checks the values of these variables and performs the necessary action if one or the other has been set.

LISTING 2 — Switch_bumper.ino.

```
#include <Servo.h>

Servo servoLeft;                // Define left servo
Servo servoRight;               // Define right servo

volatile int pbLeft = LOW;
volatile int pbRight = LOW;
boolean started = false;

void setup() {
  // Set pin modes for switches
  pinMode(2, INPUT);
  pinMode(3, INPUT);
  pinMode(4, OUTPUT);
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);      // Serves as ground connection

  servoLeft.attach(10);      // Set left servo to pin 10
  servoRight.attach(9);      // Set right servo to pin 9

  Serial.begin(9600);

  // Set up interrupts
  attachInterrupt(0, hitRight, FALLING);
  attachInterrupt(1, hitLeft, FALLING);

  started = true;
  forward();
}

void loop() {

  if (pbLeft == HIGH) {            // If left bumper hit
    reverse();
    delay(500);
    turnRight();
    delay(1500);
    forward();
    pbLeft = LOW;
    Serial.println("pbLeft");
  }

  if (pbRight == HIGH) {           // If right bumper hit
    reverse();
    delay(500);
    turnLeft();
    delay(1500);
    forward();
    pbRight = LOW;
    Serial.println("pbRight");
  }

}

// Interrupt handlers
void hitLeft() {
  if (started)
    pbLeft = HIGH;
}
void hitRight() {
  if (started)
    pbRight = HIGH;
}

// Motion routines from Listing 1 here
```

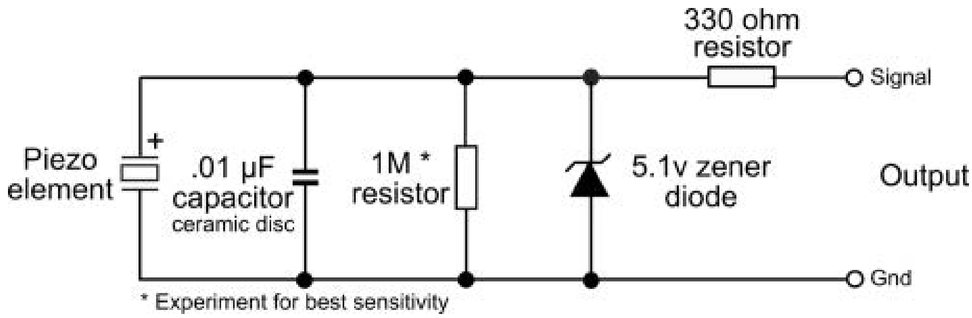


FIGURE 7. Suggested universal interface circuit for using piezo elements. The extra parts help protect the Arduino against over-voltages, and improve sensor reliability.

Using Piezo Sensors to Detect Contact

Recall from last month, we demonstrated using piezo discs and film as a way to register touch. Both types of piezo materials are commonly available and affordable — just a dollar or two.

Piezo sensors react to instantaneous pressure — either applied or released. They produce a short-lived voltage that can be registered using the Arduino's input pins. The six analog inputs are ideally suited for reading piezo sensors, as you can set a numeric threshold to determine what voltage values you wish to ignore, and what you want to consider as a hard thump against some intervening object.

Figure 7 shows a unified circuit for both piezo discs and piezo film. It incorporates a 1 MΩ resistor to bleed off excess charge developed by the sensor; a 5.1 volt zener diode to limit the voltage output; a 0.01 µF disc capacitor to help slow down the transient voltage produced by the sensor; and a 330 Ω resistor to limit current going to the Arduino.

Truth be told, some — or even all! — of these components may not be necessary. They're included either for added safety to help product the Arduino's sensitive input pins, or for improving the reliability of the readings obtained from the piezo elements.

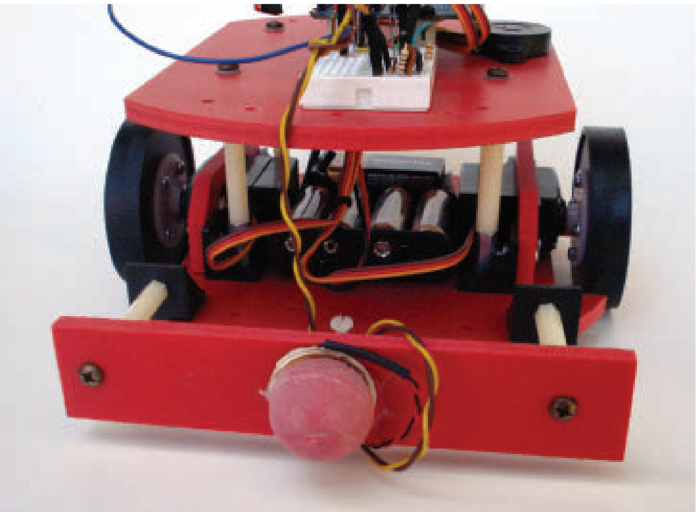
To use a piezo disc or piece of film as a contact sensor, it must be attached to your robot in some way. **Figure 8** shows a workable but somewhat hokey arrangement of gluing a sawed off rubber ball to a piezo disc, and then attaching the disc to a front bumper. (Don't laugh, it works!) Whenever the robot bumps into something, contact is registered as a voltage "blip" on the Arduino's analog A0 input pin. Given information that it's run into something, the robot can back off and turn in a new direction. To use this rig, you'll probably need to remove the two leaf switches (if you've previously added them).

To make the sensor, get a 1" rubber ball — the kind from a 25 cent gum ball machine works fine. Cut off the end of a rubber ball so that the cut surface of the ball fills about 80 to 90 percent of the diameter of the piezo disc. Use hot glue to secure the ball to the disc, then use more glue to cement the disc to a bumper that sticks out in front of the robot. I've added longer wires with male header pins to the piezo disc so that it can easily connect into the solderless breadboard atop the ArdBot. The breadboard provides ample room for the interfacing electronics.

Listing 3 provides a quick test sketch for verifying operation of the piezo disc. Upload it, then open the Serial Monitor window. The value shown represents zero to five volts, though the scale is from zero to 1023. The lower the number, the lower the voltage from the disc. The number should be zero or very close to it. Now, load up **Listing 4**. It provides code to monitor the voltage received on pin A0. If the value is above a certain threshold (in this case, 30), then the robot considers that it's bumped something. The motors momentarily reverse, then spin the robot into a new heading before going off again.

You can adjust the sensitivity of the piezo bump detector by changing the threshold value in the if statement

FIGURE 8. Touch bumper made by gluing a sawed off rubber ball to the face of a piezo disc. Contact against the ball makes the piezo change voltage which, in turn, registers on the Arduino.



LISTING 3 — Test_adc.ino.

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay (50);
}
```

Sources

All Electronics
Leaf switches
www.allelectronics.com

Budget Robotics
ArdBot and ArdBot II chassis kit
www.budgetrobotics.com

Parallax
Piezo film (605-00004)
Three-pin cable extension, 12" (800-00120)
www.parallax.com

Pololu
Female-female pre-crimped wiring (#1800)
www.pololu.com

SparkFun
Resistive sensor (SEN-09376)
Piezo vibration sensor (SEN-09196)
Piezo disc element (SEN-10293)
www.sparkfun.com

LISTING 4 — Touch_input.ino.

```
#include <Servo.h>

Servo servoLeft;            // Define left servo
Servo servoRight;           // Define right servo

void setup() {
  Serial.begin(9600);
  servoLeft.attach(10);     // Set left servo to pin 10
  servoRight.attach(9);     // Set right servo to pin 9
  forward();
}

void loop() {               // Loop through motion tests
  forward();
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  if(sensorValue > 30) {
    reverse();
    delay(1000);
    turnRight();
    delay(500);
  }
  delay(20);
}

// Motion routines from Listing 1 here
```

from 30 to something else. Use a lower value to catch even the slightest of nudges; use a higher value to reduce false triggers. Depending on the type of piezo disc you use, the weight of the ball, and other factors, if the threshold is too low a bump may trigger simply from the vibration of the robot rolling over the floor.

Another approach to bump detection via piezo elements is shown in **Figure 9**. This technique uses a small piezo film tab as the active sensor. This type of piezo sensor is available at Parallax and SparkFun; see the **Sources** box for details. The film is attached to a piezo of thin plastic strip — the kind used in report covers works well. Just cut a length of strip to about 3/4" wide by about 8" long. Secure the strip to the front of the robot, making a U-shaped loop at the front of the bumper.

When the robot makes contact with an object, the loop is pushed in, causing the piezo film to deform. As this occurs, it will generate a small voltage (about 0.5 to one volt, depending on the velocity of contact) — enough for the Arduino to pick up the change on its analog input pin.

You'll want to experiment with the size of the loop for best sensitivity. When constructing the sensor, use flexible wire with female crimp connectors on each end. You can get these wires at Pololu, among other sources. Carefully solder the connectors onto the piezo film. Be sure not to overheat! Then, use double-long male connector pins to attach the other end of the wire into the interface circuit on the solderless breadboard.

You can use the same sketch in **Listing 4**. You may need to adjust the threshold value to change the sensitivity of the sensor. You can visually check the value from the piezo sensor by opening the Serial Monitor window. Press, then release the bumper, and watch the numbers change. (If the numbers change too rapidly for you to see, increase the delay at the bottom of the loop function. When done with the test, return the value to what it was. If there's too long of a delay, the Arduino may miss some of your robot's bumps.)

You may notice that periodically the values reported in the Serial Monitor window suddenly rise, then fall again, all without you touching the sensor. This can occur when using certain Arduino functions like driving servos while interfacing to sensors like piezo elements that have a high impedance. The effect is mitigated — in part — by using the 1 KΩ series resistors between the Arduino and servo inputs. When testing either type of piezo element, be sure the

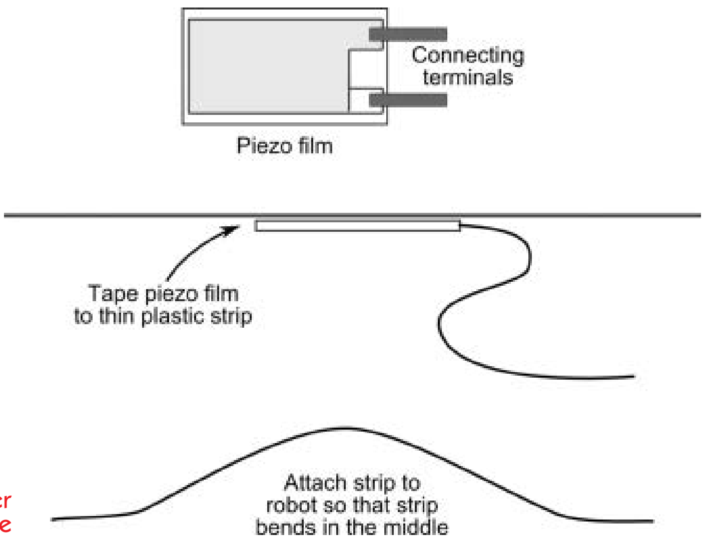


FIGURE 9. Construction details for building a thumper-bumper out of piezo film and a thin strip of plastic. Attach the strip to the front of the robot so that there's a loop.

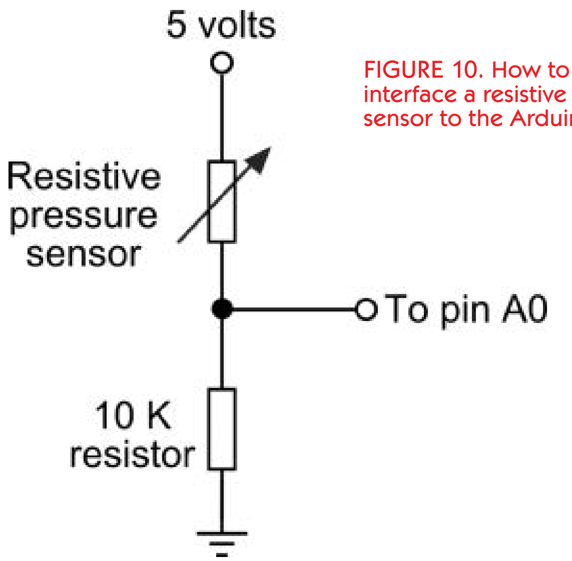


FIGURE 10. How to interface a resistive sensor to the Arduino.

servos are powered, or else disconnect the servo lines from pins D9 and D10.

Detecting Touch With Resistive Film

A third simple and cheap way of directly detecting contact is using resistive film sensors. These are often referred to by their trade name for Force Sensitive Resistor (or FSR). These sensors are essentially variable resistors that react to changes in applied pressure. They're easy to interface to the Arduino as shown in **Figure 10**. Select the fixed resistor based on the sensor you are using and the desired sensitivity. I used a 10 KΩ resistor because I had it handy. Feel free to experiment.

You can use the **Listing 4** sketch for reading values from the resistive film sensor. The range of return values is larger (and more predictable) than when using piezo elements, so you can reliably set a higher threshold. In my prototype, I got values of 700+ when applying light to medium pressure to the 1.5" square resistive sensor from SparkFun (item SEN-09376). When used as a contact sensor, the device shows contact with values as low as 50 to 100. Use double-sided tape to mount the resistive film on a bumper that extends from the front of the robot.

Last Points to Touch On

Of course, there are even more sensors your robot can use to detect when it's banged against something. Some are more elaborate than others. An accelerometer can detect the shock of impact, and can even determine when the robot has stopped when it should still be traveling. Keep your eyes open and ears to the ground, and you're sure to discover all kinds of nifty techniques to help your robot get a feel for its environment. **SV**

FIRST Robotics Las Vegas Regional Competition 2012

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

by Dave Graham

www.servomagazine.com/index.php?/magazine/article/august2012_Graham

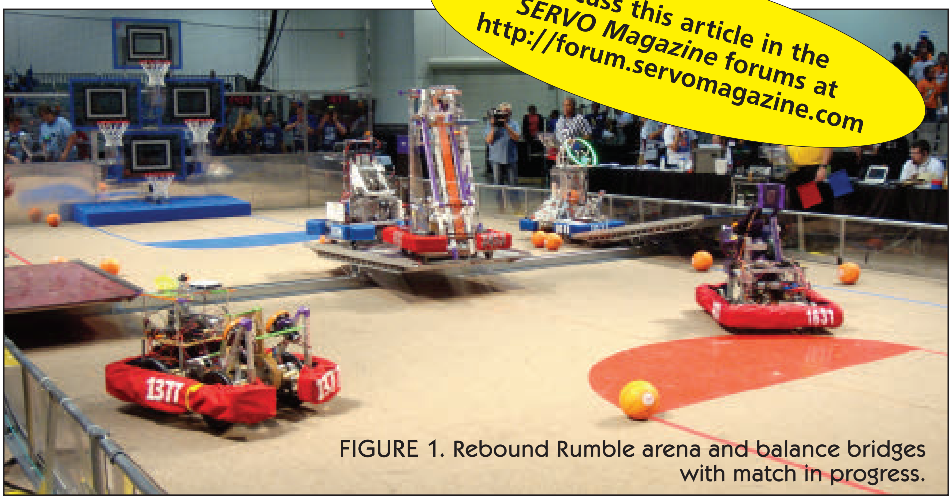


FIGURE 1. Rebound Rumble arena and balance bridges with match in progress.

The Cashman Center in Las Vegas, NV was absolutely alive with energy as 42 high school student teams from Arizona, California, Colorado, Nevada, Utah, Wyoming, and Mexico came to town the first weekend in April for a high tech shootout in the city of glitter. The contest would involve basketball-shooting robots competing in the FIRST (For Inspiration and Recognition of Science and Technology) Robotic Competition event known as Rebound Rumble.

Rebound Rumble is played on a 27 x 54 foot arena with various height baskets at each end. The middle of the arena has three balance bridges (**Figure 1**). Teams compete in randomly selected three-robot alliances. Matches are two minutes and 15 seconds long. The match begins with a 15 second hybrid period during which robots operate autonomously. However, one robot from each alliance may be controlled using a Microsoft Kinect during this period. Robots are controlled by operators for the remainder of the match (**Figure 2**) with the goal being to score as many baskets for your alliance as possible.

There is a scoring matrix that awards weighted points for baskets scored during the hybrid period and for the height of the basket. Alliances may win additional points for balancing on the bridges at center court towards the end of the match. The competition continues with the

FIGURE 2. Competitors controlling their robots from behind a Plexiglas wall.

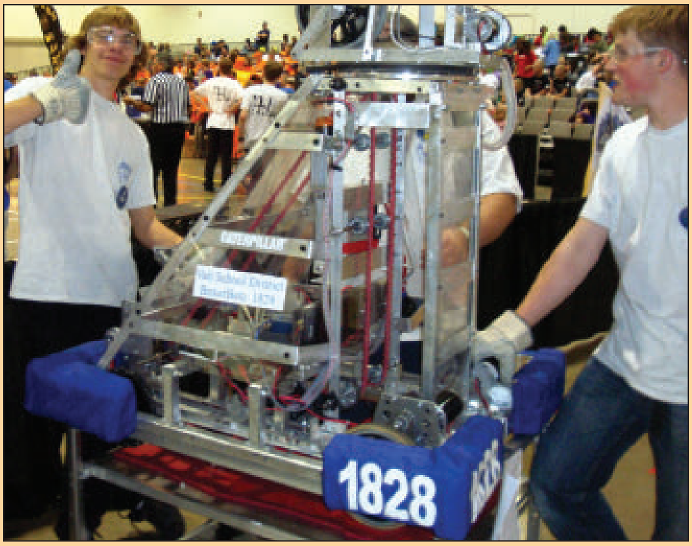


FIGURE 3. Members of Team 1828 from Vail, AZ waiting to enter the arena.

FIGURE 4. Members of Team 2844 from Laveen, AZ dressed to score tutu points.

FIGURE 6. Team mascot hugging young spectator.

FIGURE 5. Members of Team 585 from Tehachapi, CA sporting green hair.

FIGURE 7. Pit area for Team 3019 from Scottsdale, AZ.

FIGURE 8. CAD display for Team 2485 from San Diego, CA.

FIGURE 9. Warning sign on Team 3716 robot from Mount Pleasant, UT.

FIGURE 10. Part of the author's collection of FIRST team button mementos.

randomly selected alliances receiving qualification points based on a win, loss, or tie. Eventually, teams are ranked and eight alliance captains are selected based on their team rank. Team captains then pick teams as their alliance partners for the elimination matches. Alliances winning two matches advance.

This was my first FIRST competition and it was absolutely over the top. I am at a loss to describe the intensity of the energy at the venue. Action in the arena was fast paced. I was really impressed with the organization and speed at which the competitors moved their robots into and out of the arena. It appeared to be choreographed, non-stop action of robots moving to the beat of the background music at the direction of the emcees. Team spirit was over the top as well, from teams waiting to compete (**Figure 3**) to teams donning costumes, (**Figure 4** and **Figure 5**) to team mascots exciting the crowd and hugging the younger spectators (**Figure 6**). The pit area was equally alive and impressive (**Figure 7**). Many teams had CAD displays and animations of their bots (**Figure 8**) while some displayed warnings to the spectators (**Figure 9**). All competitors were obviously proud of their creations and were eager to give you a memento of their team (**Figure 10**).

I'm hooked on FIRST and encourage you to attend an event near you! **SV**

The *SERVO* Webstore

Attention Subscribers ask about your discount on prices marked with an *

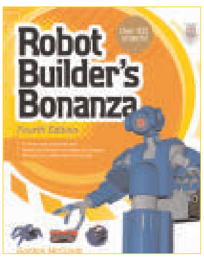
CD-ROM SPECIALS

NEW RELEASE

ROBOTICS

Robot Builder's Bonanza, Fourth Edition

by Gordon McComb



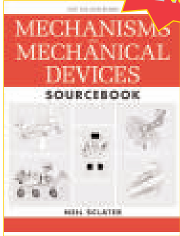
Robot Builder's Bonanza, Fourth Edition includes step-by-step plans for a number of motorized platforms. The book is described as a compendium of robotics topics, containing more than 100 projects, including 10 robot designs new to the fourth edition. These modular robots are low cost, and are made to be reproduced by readers with no training in mechanical construction.

$29.95*

Mechanisms and Mechanical Devices Sourcebook 5th Edition

NEW!

by Neil Sclater



Fully revised throughout, this abundantly illustrated reference describes proven mechanisms and mechanical devices. Each illustration represents a design concept that can easily be recycled for use in new or modified mechanical, electromechanical, or mechatronic products. Tutorials on the basics of mechanisms and motion control systems introduce you to those subjects or act as a refresher.

Reg $89.95 Sale Price $79.95

Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists

by Dustyn Roberts

In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.

$29.95*

Build Your Own Humanoid Robots

by Karl Williams

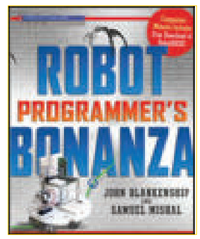
GREAT 'DROIDS, INDEED!

This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot.

$24.95*

Robot Programmer's Bonanza

by John Blankenship, Samuel Mishal



The first hands-on programming guide for today's robot hobbyist!

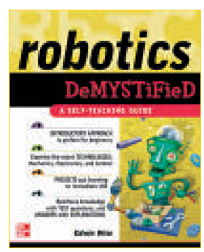
Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.

$29.95

Robotics Demystified

by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!



Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

To order call 1-800-783-4624

SERVO Magazine Bundles

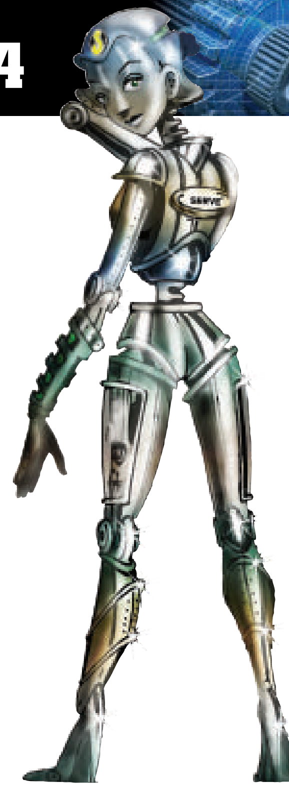
Save $10.00
Only $57.95!

Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, 09, 10 and 2011.

Any bot builders out there?
Get cool robotics stuff from my store!

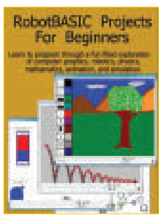
Call me at my order desk!

Visit my online store @
www.servomagazine.com



RobotBASIC Projects For Beginners

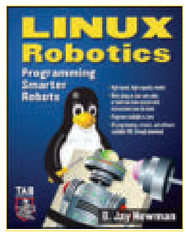
by John Blankenship, Samuel Mishal



If you want to learn how to program, this is the book for you. Most texts on programming offer dry, boring examples that are difficult to follow. In this book, a wide variety of interesting and relevant subjects are explored using a problem-solving methodology that develops logical thinking skills while making learning fun. RobotBASIC is an easy-to-use computer language available for any Windows-based PC and is used throughout the text.
Price $14.95

Linux Robotics

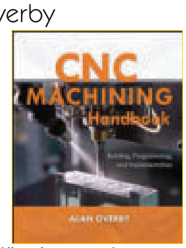
by D. Jay Newman



If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.
$34.95

CNC Machining Handbook: Building, Programming, and Implementation

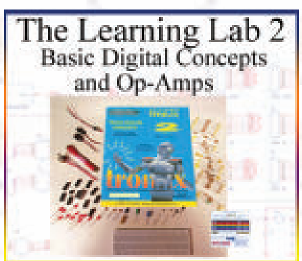
by Alan Overby



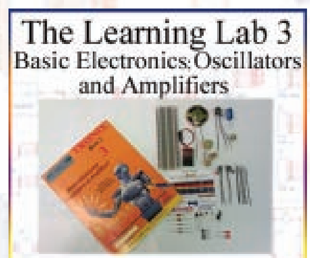
The *CNC Machining Handbook* describes the steps involved in building a CNC machine and successfully implementing it in a real world application. Helpful photos and illustrations are featured throughout. Whether you're a student, hobbyist, or business owner looking to move from a manual manufacturing process to the accuracy and repeatability of what CNC has to offer, you'll benefit from the in-depth information in this comprehensive resource. **$34.95**

FOR BEGINNER BOT BUILDERS

$59.95



$49.95



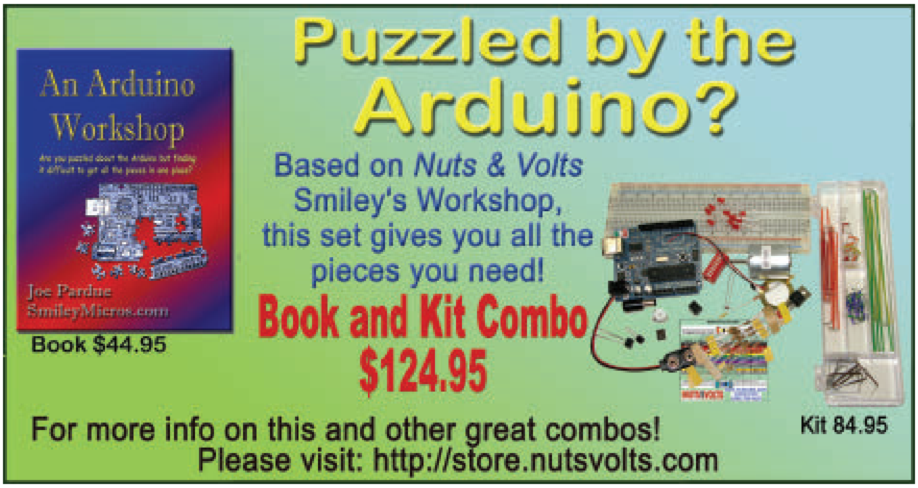
$39.95

The labs in this series — from GSS Tech Ed — show simple and interesting experiments and lessons, all done on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal of knowledge with each successive experiment.

For more info and a promotional video, please visit our webstore.

Or order online www.servomagazine.com

SPECIAL OFFERS



Forbidden LEGO

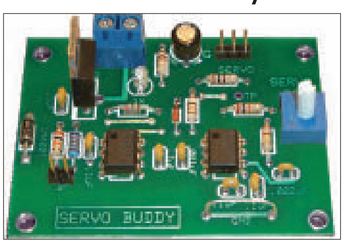
by Ulrik Pilegaard / Mike Dooley

Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse.

Reg $24.95 Sale Price $19.95

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.

For more information, please check out the May 2008 issue or go to the SERVO webstore.

Includes an article reprint.

Subscriber's Price **$39.55**

Non-Subscriber's Price **$43.95**

3D LED Cube Kit

From the article "Build the 3D LED Matrix Cube" as seen in the August 2011 issue of *Nuts & Volts Magazine*.

This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes.

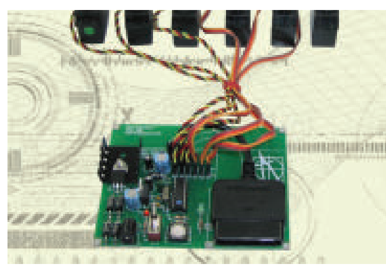
Colors available: Green, Red, Yellow & Blue.

Jig and plastic cases also available.

Subscriber's Price **$57.95**

Non-Subscriber's Price **$59.95**

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors. Includes all components and instruction manual.

For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

Subscriber's Price **$79.95**

Non-Subscriber's Price **$84.95**

Virtual Sensors

Part 2 - Virtual Motor Interfaces

by John Blankenship and Samuel Mishal

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com
www.servomagazine.com/index.php?/magazine/article/august2012_Blankenship

Some robot controllers have an integrated motor controller capable of powering small DC motors. When projects require servomotors or large DC motors though, most people assume the onboard controller is a waste — as you will see, it does not have to be.

Last month, we examined how the virtual sensor techniques we developed for our RobotBASIC Robot Operating System (RROS) can be utilized in non-RobotBASIC projects. This month, we want to do the same for virtual motor interfaces.

When we began development of our RROS, we wanted it to be able to manage the robot's drive system no matter what type of motors were used. Unfortunately, the hardware being used did not have enough I/O pins to handle all three of the motor options we wished to support. We also knew that any solution we came up with had to make each of the motor interfaces indistinguishable from each other to other modules in the RROS. We hope a summary of how we resolved our problems can benefit others in similar situations.

Some robot controllers — such as Pololu's Baby Orangutan — contain integrated hardware to directly control low current DC motors. When your robot is small enough to use such motors, this feature can be very cost-effective. Hobbyists that wish to use larger DC motors or even continuous-rotation servomotors to power their robot generally must use other I/O pins to create a serial interface to drive an external controller for larger DC motors, or to pulse the control lines of their servomotors. It would be great if the pins associated with the onboard motor controller could be used for these purposes, but usually the pins interfaced with the controller are dedicated and not made available to the user.

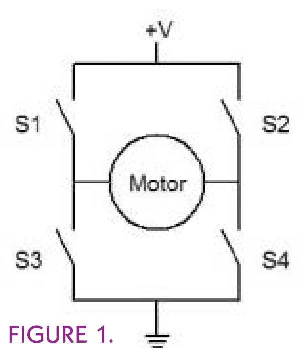


FIGURE 1.

Since the onboard motor controller used on most robot controllers is typically a commercial IC, most hobbyists assume that it can only be used for its designed purpose. Our RROS needed every available I/O pin, so we had to find ways to use the output from a DC motor controller to control servomotors and to communicate with external motor controllers.

We will use Pololu's Baby Orangutan to illustrate our points. It uses an ATmega328P as its core processor and a TB6612FNG motor controller, enabling small DC motors to be powered without additional hardware. In general, a motor controller is just four transistors acting as switches that are connected in an H-bridge as shown in **Figure 1**. If we turn on switches (transistors) S1 and S4, the motor turns one direction. Closing S2 and S3 reverses the current through the motor and turns the motor in the opposite direction. Both motor leads are shorted to ground if S3 and S4 are closed, creating a brake condition due to the motor's back EMF. If all switches are open, the motor will coast because both motor leads are effectively unconnected. The nice thing about using a controller such as the TB6612FNG is that it is easy to select the various states as described above. For purposes of this article, we will assume we have a Motor() function that can produce all of these states by passing it one of these parameters: FORWARD, REVERSE, BRAKE, or COAST.

Small DC Motors

In order to use the motor controller to power a small

```
SmallDC(dir, speed)
  {
  if(speed==0) return;
  if(dir==1)
    Motor(FORWARD);
  else
    Motor(REVERSE);
  for(int c=0;c<=100;c++)
    {
    if(c>speed)
      Motor(BRAKE);
    delay_us(100) // delay 100 micro sec
    }
  }
```

FIGURE 2.

DC motor as it was intended, we turn the motor on in forward or reverse, or place it in the brake condition to stop it. We can produce different speeds by pulsing the ON states with different duty cycles. These capabilities are easily accomplished by the function in **Figure 2**. A value of 1 for `dir` produces forward movement, while -1 indicates reverse. The speed parameter is a percentage (0-100) of the maximum speed.

Note that the `SmallDC()` function produces only one pulse, so it must be called continuously to produce the desired movement. (More on this shortly.) The 100 microsecond delay in the for-loop ensures that the total time spent in the loop is 10,000 microseconds (100*100). The motor is always turned ON in the proper direction before the loop starts, and then is turned off when the proper percentage of on-time is reached. If this routine is called continuously, the motor will be pulsed 100 times per second creating a smooth rotation.

Servomotors

The techniques shown for small DC motors are well understood by most seasoned hobbyists, but let's see how we can use this same hardware to pulse a servomotor.

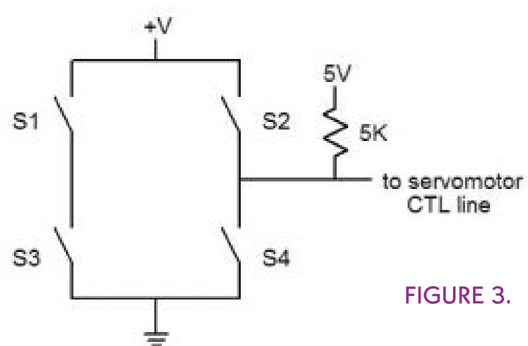


FIGURE 3.

```
Servomotor(dir, speed)
  {
  int pw=1500+dir*speed*5;
  Motor(COAST);
  delay_us(pw);
  Motor(BRAKE);
  delay_us(20000-pw);
  }
```

FIGURE 4.

Servomotors are controlled by the width of a 5V pulse applied to the control line. In general, a width of 1,500 microseconds will cause the motor to stop. As the pulse increases toward 2,000 microseconds, the motor will move forward faster and faster. Decreasing the pulse toward 1,000 microseconds also moves the motor faster, but in reverse. The next step is to see how we can produce 5V pulses with the H-bridge circuitry.

It is very important that we do NOT use the ON condition of the existing hardware because the voltage used to power the H-bridge — and thus the motors — is usually larger than 5V. We can, however, use the COAST state along with a pull-up resistor to produce a five volt pulse for controlling a servomotor; this is shown in **Figure 3**. Notice we are using the right motor-lead connection to drive the servomotor, but either motor-lead will work fine as both will react identically to our servomotor software.

The software needed to produce the appropriate pulse for a servomotor is shown in **Figure 4**. It starts by calculating the necessary pulse width (`pw`) by effectively adding or subtracting up to 500 (based on the current value of speed) from the off-time of 1,500. When the motor is set to COAST, the output will go high to 5V and stay there for the time calculated. The BRAKE condition terminates the pulse by bringing the output line to zero. Finally, a delay is used to ensure that the entire process takes 20 ms which will produce 50 pulses per second when this function is called continuously.

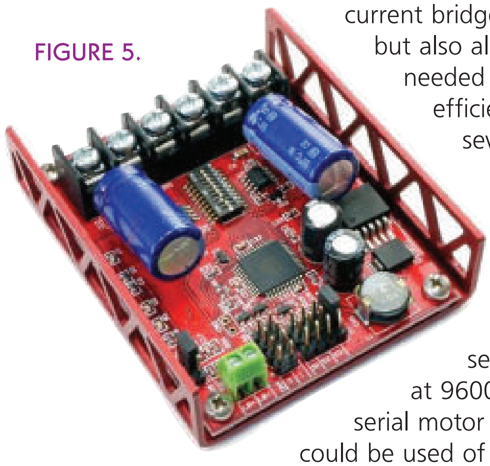
It is worth mentioning that some continuous-rotation servomotors we have used reach their peak speeds at pulse widths well inside the normal limits of 1,000 and 2,000 microseconds. This is easily corrected by reducing the multiplier of five in the first line of the `Servomotor()` function in **Figure 4**. Just experiment to find what is right for your situation.

Large DC Motors

The TB6612FNG motor controller on the Baby Orangutan can only handle enough current for small motors. This is true for nearly all IC controllers because of an IC's limited ability to dissipate heat. In order to power larger motors, we would need an H-bridge built with high current MOSFET transistors. The RoboClaw motor controller from Basic Micro shown in **Figure 5** not only has a high

The soon to be released RobotBASIC Robot Operating System is designed to make building a robot easier than ever. It provides both the hardware connections and the software drivers for interfacing a wide variety of motors and sensors with RobotBASIC. The interface itself is just the beginning though, because, as with any operating system, the RROS truly manages the resources being controlled. For example, it can automatically interface with and control all the motor types described in this article using ramping, wheel encoders, and much more. If you are curious about the features of our RROS, a draft of the User's Manual can be downloaded from **www.RobotBASIC.com**.

FIGURE 5.



current bridge (30 amp), but also all the firmware needed to drive it efficiently using several types of control signals. For our example, we will assume it has been set up to receive serial commands at 9600 baud. Other serial motor controllers could be used of course, but the programming example provided here is compatible with RoboClaw controllers.

The RoboClaw can control the speed and direction of the motor connected to it based on the value of a single byte sent to it over the serial communication link. The expected serial signal should vary from zero to five volts, so we can use the circuit from **Figure 3**. Just connect the integrated motor controller's output to the serial input on the RoboClaw instead of to a servomotor.

Figure 6 shows a function for controlling motors via a RoboClaw controller. It produces a single byte of data based on the `dir` and `speed` provided, and sends that byte serially to the RoboClaw controller by manipulating the output lines on the TB6612FNG controller so that it appears to be a serial output port.

A byte value of 64 tells the RoboClaw to stop the motor. The RoboClaw can actually control two motors, but to keep things simple our example will only deal with one. As the byte value increases toward 127, the motor will speed up in one direction. Decreasing the value toward zero will increase the motor's speed in the opposite direction.

The actual time duration for each bit at 9600 baud should be 104 microseconds. Our code used a period of 100 microseconds because the other statements in the loop have delays of their own. You may have to alter the timing slightly, depending on the processor you are using.

Virtual Controllers

Using our software, the TB6612FNG motor controller can now control both large and small DC motors, as well as servomotors. Don't forget that the `SmallDC()` and `Servomotor()` routines must be called continuously in order to properly activate the motors. Since it does not hurt to continuously call the RoboClaw routine, we can treat it exactly the same in an application.

The simplified code in **Figure 7** demonstrates how all three motor drivers can be integrated into a single program that includes reading sensors, and calculating new motor speeds and directions based on those readings. This structure allows the same program to work — without modification — with any selected motor type.

Enhancements

Of course, each of the motor routines discussed here can be enhanced as you see fit. For example, for most robot projects they would have to be modified to handle two motors. Once you understand the principles involved though, adding another motor is just more of the same. You could also embellish the routines by adding code to make the motors automatically ramp up and down slowly when speeds are changed. You could even add routines to monitor and utilize wheel encoders. These are only some of the features integrated into our RROS.

As you can see, creating virtual interfaces for various motor types lets you control all three types without the need for extra I/O pins. These techniques also make it possible to create one general-purpose motor control driver that you can use in all your projects, no matter what type of motor you are using. SV

```
RoboClaw(dir, speed)
  {
  int d = 64+(speed*Dir*64)/100;
  int i;
  int BitTime=100;
  Motor(BRAKE);            // start bit
  delay_us(BitTime);
  for(i=0;i<8;i++)
    {
    if(d&1)
      RightMotor(COAST); // send a one
    else
      RightMotor(BRAKE); // send a zero
    d=d>>1;
    delay_us(BitTime);
    }
  RightMotor(COAST);      // stop bit
  delay_us(BitTime);
  }
```

FIGURE 6.

```
#define DC 1
#define Servo 2
#define Claw 3
// set the motor type you are using
main()
  {
  int MotorType = Servo;
  while true
    {
    // read any sensors here
    // use sensor data to calculate
    // new speed and dir
    switch(MotorType)
      {
      case DC:
        SmallDC(dir, speed);
      break;
      case Servo:
        Servomotor(dir, speed);
      break;
      case Claw:
        RoboClaw(dir, speed);
      break;
      }
    }
  }
```

FIGURE 7.

Easy To Build Wireless Servo Controller

by Robert H. Walker

www.servomagazine.com/index.php?/magazine/article/august2012_Walker

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

If you are in need of a simple, inexpensive, and easy to construct wireless remote control that can independently operate up to two servomechanisms, this design may be just the answer for you. Wireless remote controls for servomechanisms have always been an integral part of R/C projects (including model airplanes and model cars). The numerous available systems can handle many channels and are very elegant, but in general are also quite costly. The total cost of parts for this project is less than $50.

There are only three key components employed in this design, as shown in the circuit diagram in **Figure 1**. The first is a 27 MHz or 49 MHz toy car two-channel receiver which has been removed from the car chassis. The second component is a micro-sized gearbox which is linked via shrink sleeving to a potentiometer. The third part is a 555 IC timer based servo controller.

Figure 2 is a top down view of the controller module. Starting with the receiver board at the top, go clockwise around to the gearbox/potentiometer linked pair, and then to the servo controller board. The only other part is the gearbox speed adjustment — a Cermet trimpot. The entire module measures 5" x 3" and could obviously be made a bit smaller, since many toy car receivers are actually much smaller than the one shown.

The gearbox is a Microgear Motor Block (45 rpm) available from **ServoCity.com**. (An alternative is to modify a small servo to operate as a gearbox). The servo controller is available as an easy to assemble kit from **HansenHobbies.com**. The only revision necessary is to replace the included board mounted 10K trimpot with an outboard wired 5/8" diameter 10K linear taper potentiometer. The potentiometer has to be mounted with the shaft horizontal and lined up with the output shaft of the gearbox.

The heat-sensitive shrink sleeving is slipped over both shafts and then shrunk down. The sleeving links the gearbox to the 10K pot and will also act as a slip clutch when the pot hits its

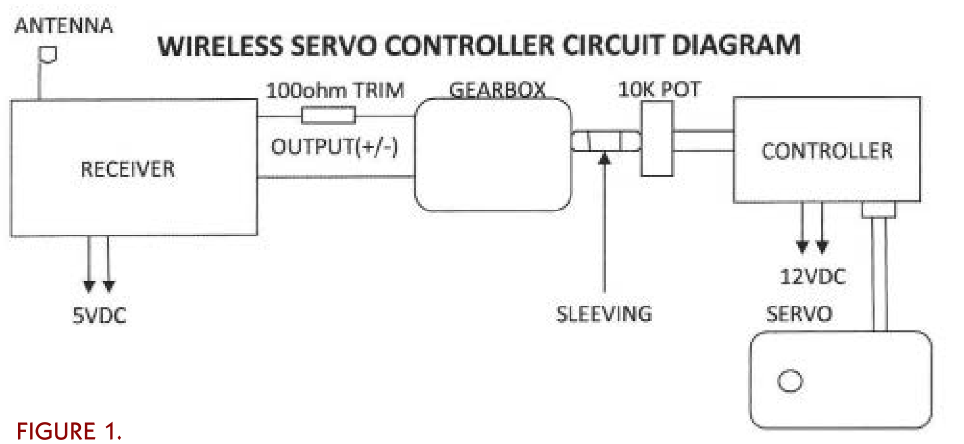


FIGURE 1.

end stops. The speed of the servo movement can be adjusted with the 100 ohm Cermet trimpot which is wired in series with the receiver output to the gearbox.

An RF signal sent to the receiver from its matching transmitter will result in either a positive or negative 4.5 VDC output, depending on which way the transmitter key is pushed. This will cause the gearbox to rotate either clockwise or counterclockwise which will also rotate the 10K pot. The pot will then change the controller pulse chain which will signal the servo to move. The servo will continue to move as long as the transmitter key is pushed, until it hits its endpoint.

Pushing the transmitter key in the opposite direction will reverse all of these actions and cause the servo to move in the alternate direction — again, until it hits its endpoint. A small servo is shown in **Figure 2**, but any sized servo can be controlled depending on how much 12V power is available. Effective range with these transmitter/receiver combos can reach up to 75 feet.

The proper input voltage for the receiver is 5 VDC. The servo controller requires 12 VDC. You could obtain the receiver voltage by means of a +5V fixed voltage regulator such as the NTE960, fed from the 12 VDC supply (if it is more convenient to use a single power supply).

The circuit diagram in **Figure 1** shows only one remote controlled servo, however, most toy car transmitter/receiver pairs are equipped with two separate output channels. With a single receiver, you could independently operate two servos by adding a second gearbox/pot/controller setup which is driven by the second receiver channel.

I hope this circuit serves you well. **SV**

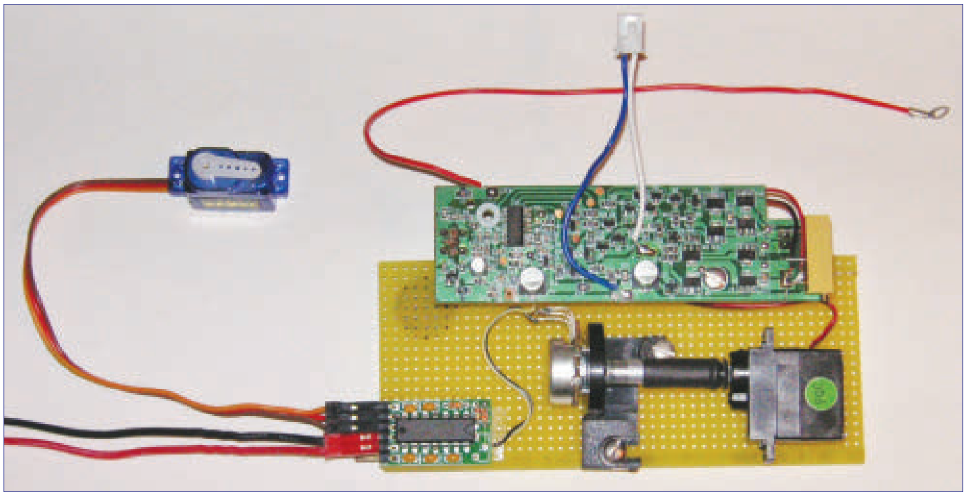


FIGURE 2. Top view of assembled wireless servo controller module showing receiver, servo control board, and gearbox linked to potentiometer.

FIGURE 3. Remote controlled auto carrier with servo actuated auto ramp tilt mechanism. Wireless controller module is in gray car to the left of the auto carrier.

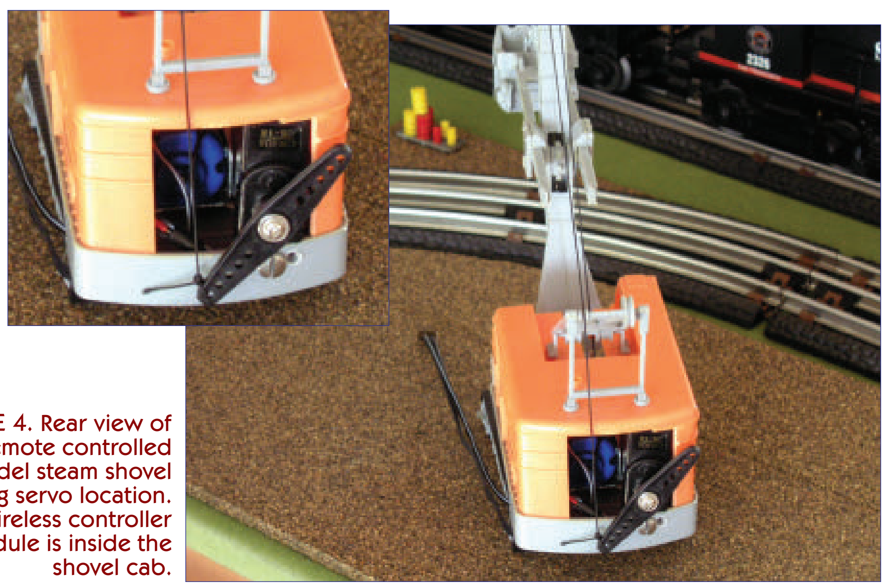


FIGURE 4. Rear view of remote controlled model steam shovel showing servo location. Wireless controller module is inside the shovel cab.

Parts List

ITEM
Receiver/transmitter pair*
100 ohm Cermet trimpot
Microgear motor block (45 rpm)
10K linear 5/8" dia potentiometer
Hansen servo controller (kit)
3/8" dia shrink sleeving
Perforated phenolic board

** Removed from toy car*

The Tale of TupperBot

By Ángel Hernández

"Yes, this is a robot and yes, it is Tupperware, but there is no room for your food!" That's how most of my conversations start when people ask me about TupperBot.

Have you ever felt like spending hundreds of hours trying to figure out how to put four wheels together on a cool robot you have in mind? Have you ever spent more time struggling with the mechanics of your robot than designing the electronics or programming cool features? That's usually how I feel, and how TupperBot was born.

Why Tupperware?

I live in a flat in downtown Madrid, Spain, so I don't have enough room to have nice machinery tools, but I love making robots. I try to keep them simple, good looking, and fun. This challenging problem has an easy solution: Tupperware! Those plastic containers we use for food have a second calling as a robot base.

TupperBot 0.1: Just learning some robotics here.

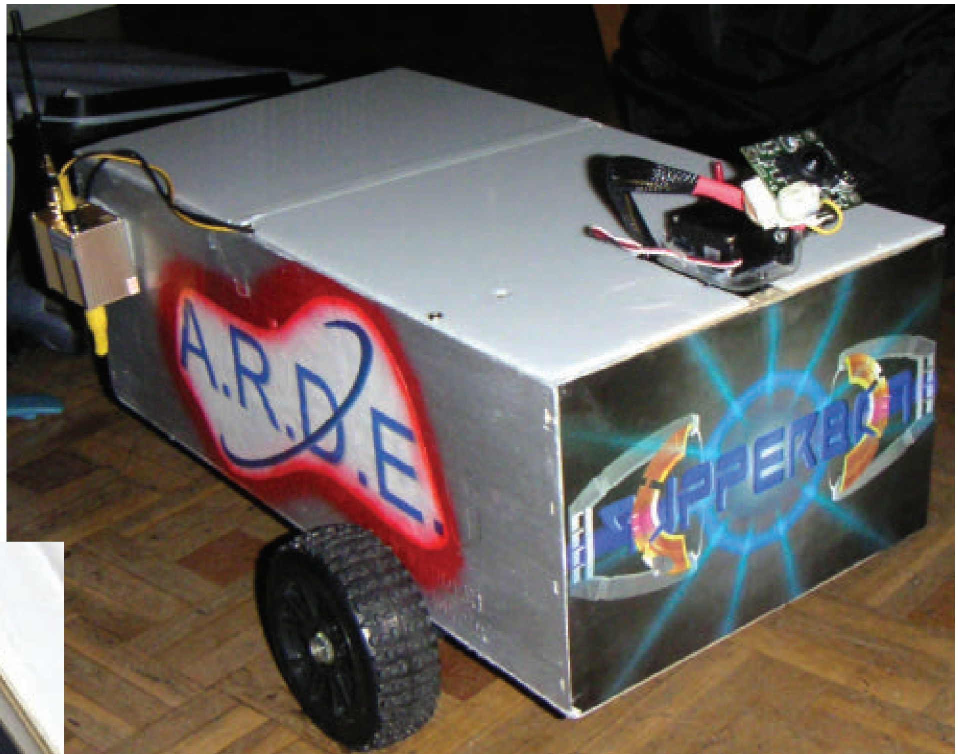
It was 2006 when I made my first robot during a large tech festival. I had just a couple of electric screwdrivers, a handmade microcontroller board, some relays, and a small plastic box where I kept some of my tools during events. It was at that point I decided it would be a good idea to put everything together to make a fun, improvised robot. However, I didn't have anywhere to keep my tools anymore.

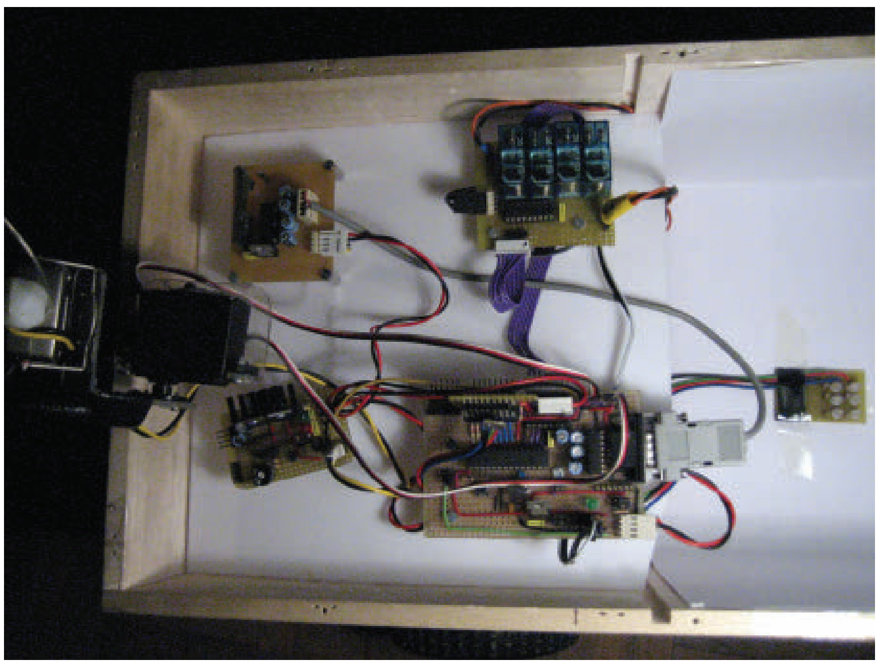
TupperBot was born!

TupperBot 1.0: First contest, first win!

Later that same year, I improved the design by adding some remote control, video, lighting functionality, and then changed the chassis from plastic to wood.

I won my first national contest at MadridBot, but wood was not that easy to work with. It took me too long to build TupperBot 1.0 compared to the first improvised TupperBot, so I decided to go back to my original idea: Tupperware.





Ángel Hernández
Twitter: @mifulapirus
Blog: **www.tupperbot.com**
Email: angel@tupperbot.com

Skypic: Microcontroller board developed by IEARobotics **www.iearobotics.com**

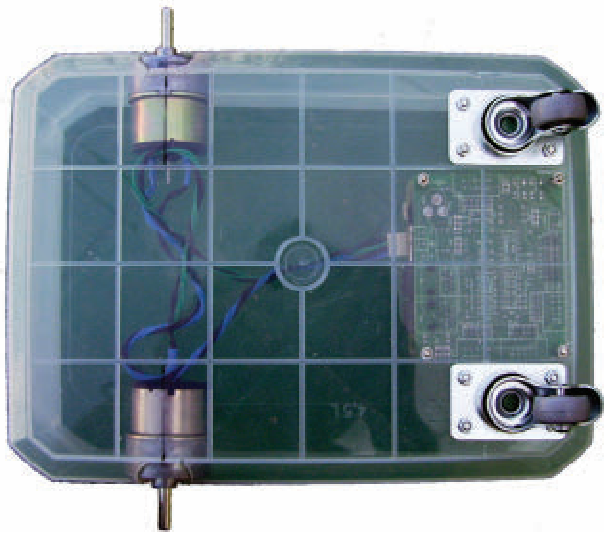
MS Agent: **www.microsoft.com/products/msagent/main.aspx**

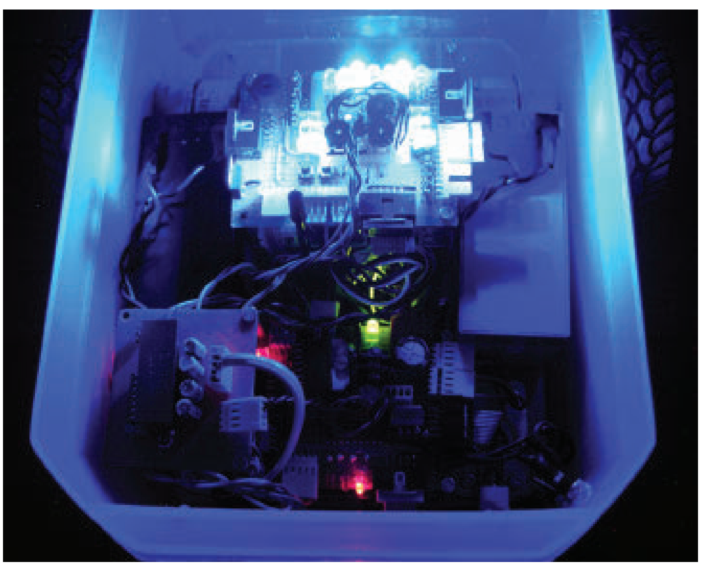
Bass.Net website: **www.un4seen.com**

Chronos eZ430: **http://processors.wiki.ti.com/index.php/EZ430-Chronos**

TupperBot 2.0: Compact and a bit more complex.

TupperBot 2.0 was again made with Tupperware, but was much more complex than the previous versions. Let me describe the internals of this robot.





TupperBot's 2.0 Hardware

All the hardware had been designed to keep it modular, so it was really easy to troubleshoot. Except for the microcontroller boards, the rest of the circuits were built using prototyping boards. I probably spent over a hundred hours building these circuits.

- **Control Towers**: Both towers use customized Skypic[i] boards with a PIC18F2525. The taller one takes care of all the main processes of the robot such as peripheral communications and driving systems. The other tower controls the servo signals, music generation, temperature, and GPS.
- **Motor Driver**: Two parallel L298s drive the four wheels in differential mode.
- **Communications**: The ER400TRS radio modem sends all the local information to the computer where the heavy calculations are done. Internally, the robot contains two I^2C and three serial ports to communicate with all the modules.
- **Sonar Sensors**: Six SRF08s let TupperBot 2.0 know about its environment so it can drive safely.
- **Sound**: One of the microcontrollers is able to play some songs using one of its PWM pins. Modulating those notes was fun!
- **Video**: I used a standard wired surveillance camera with IR lights. Video is sent to the computer through a wireless radio link.
- **RGB Lighting**: The robot is able to create any color inside and even has a "heart," making the Tupperbot 2.0 logo beat faster or slower depending on the speed of the wheels. All the PWM control of the six different lights (RGB + IR camera light + white camera light + heart) is done by a separate microcontroller embedded on the lid of the robot

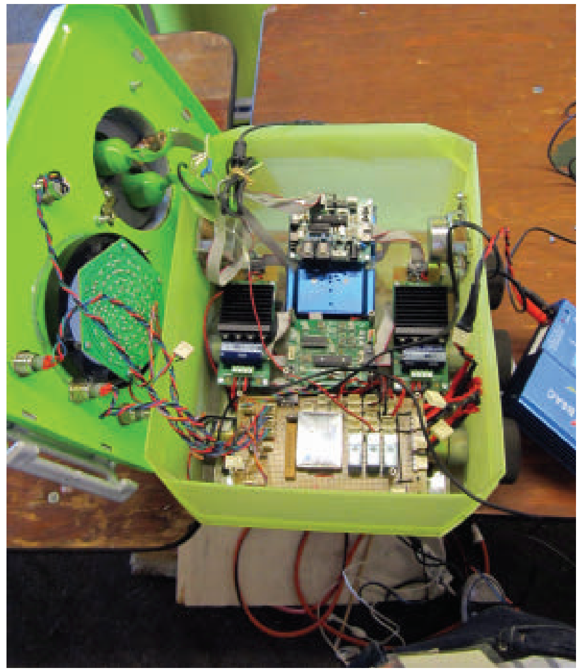
TupperBot's 2.0 Software

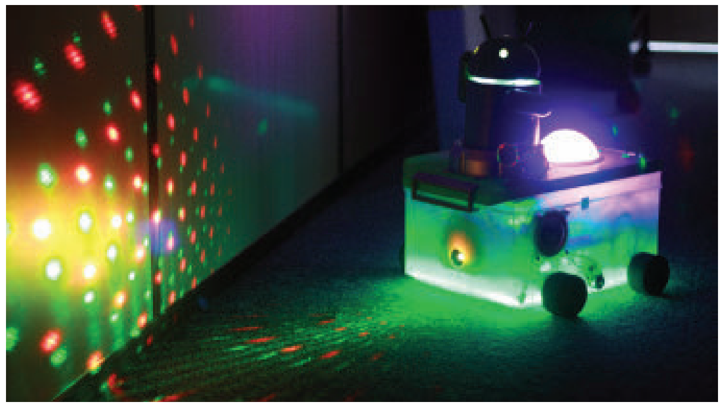
The software for this robot started off quite simple by just controlling it from a keyboard or running the robot in autonomous mode, but I started being interested in designing different interfaces for my new toy. In just a few months, I had more than 5,000 lines of code written for controlling the robot. I used C# to do all the video processing for the hand gesture's control. VB.Net was used for integrating other features such as a Wii remote control, voice recognition, text to speech, etc. I used the old MS Agent in the software. All the software is available on my blog at **www.tupperbot.com**.

I gave several talks in Spain and the US about TupperBot 2.0. It won four different contests in Spain and even a gold medal at RoboGames 2010, so it has been a very fun and productive project. However, I wanted to keep it growing. In March '11, I decided to create a whole new robot for RoboGames that same year. Of course, I only had about one month to do it in!

TupperBot 3.0: the PartyBot.

After four years of working on my TupperBot family, I decided to change gears and create a whole new robot with the same soul, but a different target: Compete in every single "Best of Show" contest I could take it to. As I stated, RoboGames 2011 was just a month away when I decided to start this new project. Google was giving some prizes there, so my brother gave me the idea of making an Android-related robot. There was no time! I was on the other side of the world and I did not even have the flight tickets to go to California ... challenge accepted! The robot had to be fun, easy to control, and have some surprising features, so here are some of the things I included in this 3.6 gallon (14L) Tupperware:





- **The Android**: Three servos control the arms and head. Five white LEDs give some light to the neck, and two sets of RGB LEDs let you control the eye color.
- **The Lid**: Four buttons let you select different songs and movements, and a full 50 RGB LED DMX spotlight adds to the party.
- **The Disco Laser**: A laser stage lighting device is sitting in the front side of the robot. I had to hack a standard one in order to control it with my own electronics.
- **Internal Lighting**: A 5W green LED illuminates the bottom of the robot. My friend Bob Allen made me a little aluminum bracket to hold it and dissipate all that heat.
- **Microcontrollers**: Three PIC16F2525s control all the lighting, communications, and servo signals in this robot. Thanks to what I learned with TupperBot 2.0 and its modular design, I could reuse most of the code.
- **The Brain**: An Asus-Eee 900 sits on top of all the electronics. I programmed a new control panel for this robot, including new features such as audio processing and cool ways to control the robot.
- **Sound Analysis**: I started doing my own Fourier transforms and sound analysis, but I was running out of time, so I found the **Bass.net** library which helped me with the math. Once I integrated the library into my code, I had all the peripherals synchronized with the beats of the music. The robot started looking like a real PartyBot!
- **Remote Control**: Moving the robot with a joystick or a keyboard is just not enough for a "Best of Show" competition, so I decided to drive it with my watch. You didn't expect that, did you?

I had a couple of Chronos eZ430s (a highly integrated, wireless development system) available at home, so I was able to get axis information from this cool gadget to control the wheels and basic functionalities of my robot.

I finally made it to RoboGames and won a silver medal in 2011, so I was pretty happy with the results of just one month of hard work. That is definitely not the best part of it, though. Since RoboGames, I have been giving talks about different aspects of the robot in several places, and TupperBot 3.0 has been "acting" in a theater called "Teatro Alfil" in Madrid, where it interacts with the audience, walks them to their seats, and introduces the play "SensorMen" — a fun comedy where four guys have MIDI sensors on their body and play really cool music.

Conclusion

Many people asked me what the purpose of that "useless wheeled plastic box" was in 2006. I struggled trying to look for a cool usage to give them a good answer, but now I realize how every single development I've accomplished in the last five years was not an end unto itself; it was just a means to learn and give me the green light to all the creativity a roboticist has inside. Every email, tweet, comment, and question I receive from people who read my tutorials and posts are the reasons I keep developing this exciting hobby.

Making robots is not what we do. It is our lifestyle. SV

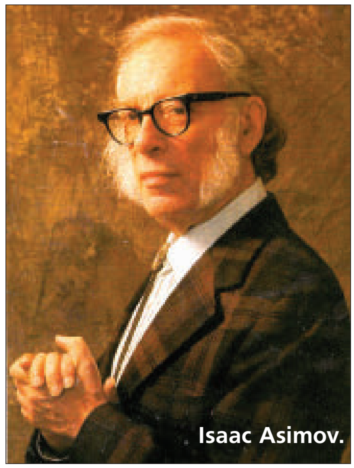
Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Then and NOW

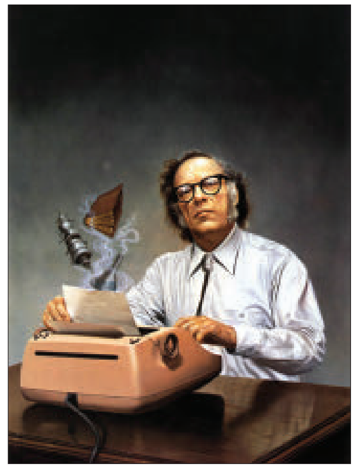
Isaac Asimov *Inspiring Today's Robots*

by Tom Carroll

For anyone who has ever read about, dreamed about, or just thought about robots, the name Isaac Asimov usually comes to mind. Dr. Asimov passed away 10 years ago this past April, and left a legacy of over 500 books published. This prolific writer has had more influence on the field and science of robotics than perhaps any other person on Earth. Everyone I know who has any interest in robots always has their particular story or novel written by Asimov that first comes to mind. The short story, *Robbie* (also known as *Strange Playfellow*), published in the compilation of stories entitled *I, Robot* (Figure 1) is the one I always think of first.



Isaac Asimov.



Robbie was a robot babysitter in a story that took place 'way into the future of 1998.' (The story was published in 1940, so set your mind on the world before WWII). Robbie was a large, metallic bipedal humanoid who was mute, but a perfect companion for little Gloria Weston — at least in her (and in her father, George Weston's) opinion. Mrs. Weston hated the machine and was determined to get rid of it. As the story progresses, Robbie is finally taken away at the demands of Mrs. Weston, but much to the chagrin of young Gloria who becomes very depressed.

The Westons finally decided that Gloria had forgotten about Robbie, so they decide to have a family outing at the Museum of Science and Industry in nearby New York City. A talking robot was on display that took up 25 square yards of wires and coils. Suddenly realizing her missing, Mrs. Weston finds Gloria near the robot display asking the robot, *"Mr. Robot, sir, have you seen Robbie?"* Mrs. Weston chastises her daughter for wandering off.

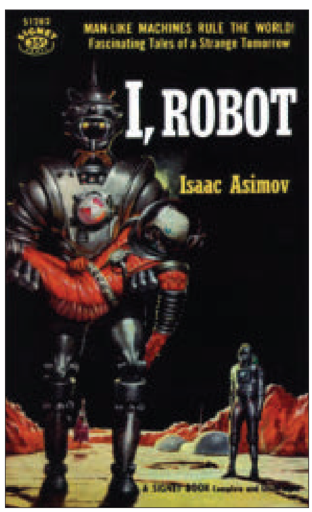


FIGURE 1. Isaac Asimov's *I, Robot* short story collection.

An important character in later Asimov stories is introduced: college student, Susan Calvin, is observing the talking robot and writing a paper entitled "Practical Aspects of Robotics." Susan will later become a robopsychologist for US Robots and Mechanical Men, Inc.

Mrs. Weston gives in a bit and finally allows the family to visit a 'staged' factory demonstration at US Robots and Mechanical Men which was set up by George, but unknown by his wife. Gloria suddenly sees Robbie on a production line and runs to him, not seeing a programmed tractor heading her way. Robbie sees the danger and runs to scoop up young Gloria just in time. Robbie was not an industrial robot and was placed there at a prior suggestion by George, though the dangerous tractor was not part of the plan. Of course, the Weston family had to keep the life-saving Robbie forever. This (and

most of Asimov's stories) depicts robots as human-friendly creations. All robots in Asimov's stories and novels adhere to the "Three Laws of Robotics," first appearing in his 1942 short story, *Runaround*, which I'll discuss later.

Asimov's Life

Isaac was born in Petrovichi, Russia on January 2, 1920 as Isaak *Judah* Ozimov (or Isaak *Yudovich* Ozimov). His parents were Orthodox Jews and the family moved to Brooklyn, NY in 1923 where his parents changed their name from Osimov to Asimov. They also changed his birth date to September 7, 1919 so he could enter school a year earlier. Always a brilliant child, he graduated from high school at age 15 in 1935. He received a B.S. in Chemistry in 1939 and a M.Sc. in Chemistry in 1941 from Columbia University. After WWII, he earned his Ph.D. in Chemistry in 1948. While in school, he tried for admission to the Boston University's School of Medicine, but was rejected. He later worked his way up to a full professor of Biochemistry, though his burgeoning writing career brought him far more money.

As all of his education was progressing, his true love continued to be writing. He eventually became the author (or co-author) of over 500 books. First writing at the tender age of 11, by 19 he had many science fiction fans. One of his stories, *Nightfall*, written in 1941 was voted by the Science Fiction Writers of America as the best short science fiction story of all time in 1964. A fictional planet is continually illuminated by six suns. However, when their alignment allows the planet to fall into night once every 2,049 years, their whole society collapses. Of course, there is far more to the story, but as is typical of Asimov writings, psychology plays a greater part than the actual setting.

Another stirring story with this "mindful" background is his 1957 tale, *Strikebreaker*. The story is about a man who is brought to a small planetoid to stop a strike by a family who runs the human waste disposal system for the whole society. The family is dearly needed to prevent waste buildup and ultimate ravages of disease, but they are 'untouchables' and cannot be communicated with due to archaic rules of handling human waste. The visitor stops the strike and he is thanked but is also now an untouchable, so therefore must leave. Both stories are not robot based, but are must-reads to understand all of Asimov's works.

Asimov and the Robots

It is the robots that we all remember Asimov for. However, as a biochemist with no education whatsoever in electronics, mechanical engineering, or computer science, his writings have long given us the feeling that he truly understood what it took to build a robot. His grasp of the early concepts of robotics was due to his high intelligence and a desire to learn about everything in the universe. His robots were almost exclusively bi-pedal humanoids. His stories arose decades before George Devol and Joe Engelberger developed the first industrial robots, so mankind was used to thinking of robots as mechanical entities similar in form to humans. From the play, *RUR*, and on, robots were humanoids.

Runaround Introduces Asimov's Robot Basics

His story, *Runaround,* was written in 1941 and published in 1942 in the pulp monthly, *Astounding Science Fiction*, and introduced the Three Laws of Robotics. The story was about two men (Gregory Powell and Mike Donovan) sent to Mercury in 2015 — 10 years after a failed mission to a selenium mine which was a source for material needed for solar power. (Far more solar-efficient silicon wafers of today were just a lab curiosity in 1941.) One of the newest robots there, Speedy, began to have problems with the three laws and started having a drunken reaction to the selenium source.

Here is the introduction of the laws in *Runaround:*

"Now, look, let's start with the three fundamental Rules of Robotics — the three rules that are built most deeply into a robot's positronic brain. We have: One, a robot may not injure a human being, or, through inaction, allow a human being to come to harm." (Powell speaking)

"Right!" (Donovan responding)

"Two," (continued Powell) *"a robot must obey the orders given it by human beings except where such orders would conflict with the First Law."*

"Right!"

"And three, a robot must protect its own existence as long as such protection does not conflict with the first or second laws."

These laws have inspired more people than any other aspect of his writings.

Positronic Brains

"No, they're robots. I've spent all day with them and I know. They've got positronic brains: primitive, of course," was another comment made by Powell in the story about

The following is a reference to Susan Calvin's early years at US Robots. She was once asked: "Are robots so different from men?" She replied, "Worlds different. Robots are essentially decent." 'All that had been done in the mid-twentieth century on "calculating machines" had been upset by <Lawrence> Robertson and his positronic brain-paths. The miles of relays and photocells had given way to the spongy globe of plantinum iridium about the size of a human brain.'

I'm not sure what plantinum is; I believe it might have been a mistake for platinum, a common addition to iridium to produce an alloy of the two. Asimov was probably just looking ahead to replace 'miles of relays and photocells' and the 'calculating machines' with this new substance.

the old robots they discovered. The positron was actually newly discovered in the time period of Asimov's early writings, and became a buzz word of the time — much like the over-used 'mega' of today to describe anything great. Asimov used the analogy of positron and electronic to form 'positronic' which sounded science-fictiony. He was always vague about the details, except that it was formed from an alloy of platinum and iridium, and contained volatile memory vulnerable to radiation.

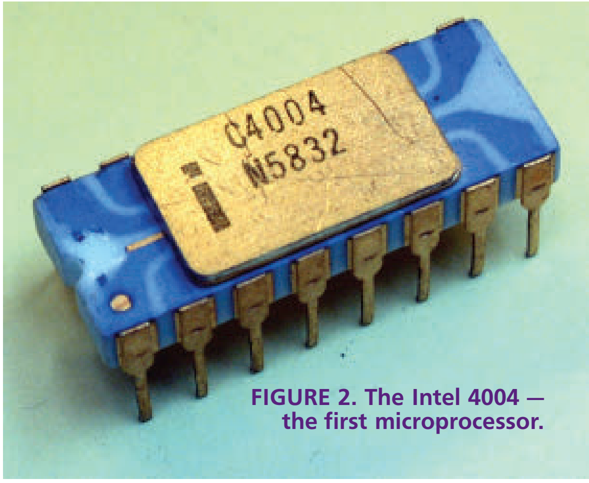


FIGURE 2. The Intel 4004 — the first microprocessor.

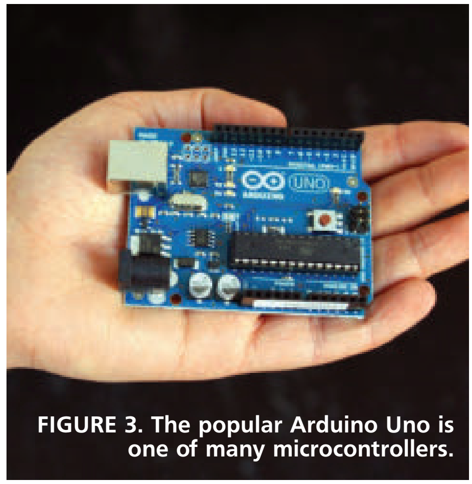


FIGURE 3. The popular Arduino Uno is one of many microcontrollers.

Over the years, people have speculated that some sort of computer brain could be made using positrons in some way, but today's silicon substrate microprocessors and microcontrollers are a far better building block on which to form a computer brain.

I remember back in early 1971 when Intel introduced the 4004 microprocessor chip shown in **Figure 2** for Busicom's advanced calculator. The four-bit chip used eight-bit instructions, ran at up to 740 kHz, and contained an amazing 2,300 transistors. Intel later brought out the 4040, 8008, and finally the 8080 and 8086 microprocessors. When these 'made the scene,' many people stated that Asimov's positronic brain had arrived. (Well, sort of.) The 6502, Z-80, 6502, and many other microprocessors followed in this computer revolution and robot builders were salivating at the many possibilities that were becoming available for their machines.

Later microprocessors have surpassed several billion transistors on a singe chip. Asimov may not have understood the technicalities of what was unfolding before him in the '70s and '80s, but he had a true appreciation of the possibilities of these devices, especially when he started using a computer for his writing and set aside his trusty typewriter.

Robots these days rarely use a microprocessor since their sensors and control systems can easily communicate with each other via simple links and languages via cheaper micro*controllers*. PICs, Arduinos (**Figure 3**), BASIC Stamps, Propeller chips, and others have taken over the control of most of our robot's functions. I am sure that today's cheapest microcontroller running the simple Basic language could run circles around the most massive computers of the early '40s. Some microcontrollers cost as little as $.25 each in quantities. If one of these had been available in 1941, Asimo would have most certainly written about it.

(Powell speaking) *"..... That's because they're supplied with the old McGuffy gears. I've been over the insides — crummiest set you've ever seen."*

Robot Movements Need Many Types of Mechanisms

As a brilliant writer who was versed in most all of the sciences — including astronomy, physics, chemistry, biology, and, of course, biochemistry — engineering was not Asimov's strong suite. In his eyes, if it moved it had to have gears, and the set of gears must have been invented by a named person. Asimov's robots on Mercury had chests 10 feet around just to contain the gears. Certainly most robots have gears — even if they are just ones in a small radio control system's servo — but they are not the key to how a robot moves.

I assume that the 'mechanical' part of Asimov's mythical company — *US Robots and Mechanical Men, Inc.* — implied his view of the importance of the mechanical aspects over electronics.

We know that gears, belts, or even chains can provide a reduction of speed from the output of an electric motor, but we usually look at the complete motor-gearbox system to determine speed and torque needs. Those of us who have constructed humanoid robots know that the placement of the different servos required to give the robot multiple axes of motion result in legs that bulge at the knees, elbows, and ankles — unlike the smoother leg shapes of humans and animals. Biologic creatures derive rotary motion via linear actuators (muscles) and not rotary mechanisms.

Industrial robot arm design has always encountered the dilemma of using either linear actuators or rotary joints. Many robot designs use both. Linear actuators can provide more lift, acting like a human muscle. For robot arm use, they are limited to a certain shoulder or elbow angle. Imagine yourself trying to curl a 135 pound barbell with your arms straight out in front of you; you'd be lucky to lift even 35 pounds. The lift becomes easier as your elbow reaches 90º as in **Figure 4**. It's the same for robot arms; rotary actuators work best in some locations and linear in other places, such as powerful robot arms.

FIGURE 4. The preacher curl method can strain both human and robot muscles. *(Photo courtesy of SkinnyBulkUp .com.)*

FIGURE 5. Today's movies require complex CGI robots.

Asimov's stories did not include any graphics, so the reader was left to imagine his or her own robot design to fit a particular scene. When movies came along, robots had to visually fit the reality of the times. Readers — particularly science fiction readers — and movie goers knew enough about basic mechanisms that set directors and prop builders had to make robots look real. 'Robot suits' worked okay at first, but then came action props that I personally had the enjoyment of building for a few movies. These props had to at least *look* as if they were working. Today's movies require such realism now that computer graphics imaging is the only way to go. Can you imagine trying to build an action prop with the mechanical complexity of the ones shown in **Figure 5** from the film *I, Robot*?

Robot Power for Asimov's Robots

Atomic energy had passed the drawing board stage in 1941 and was a working 'atomic pile' in 1942 in a squash court under an old stadium at the University of Chicago. **Figure 6** shows this reactor — it's literally a pile of blocks hiding a core. Few people really knew much about atomic power in the days of Enrico Fermi and the Manhattan Project of WWII. These days, we realize that an atomic power plant is really no different from any other type of steam power plant except in the method of creating heat. Coal, oil, or a nuclear reactor heat water to steam to drive a turbine which, in turn, drives a generator to produce electricity. The only thing that really makes a nuclear power plant different from the other types of power plants is the safety requirements for containment of the radioactive material in the reactor.

So, what was 'the two inch sphere that contained the tiny spark of atomic energy that was the robot's life?' Let's place the term 'atomic energy from a tiny sphere' in the same category as the positronic brain. It just sounded science-fictiony back in the early 1940s.

'He <Powell> had unscrewed the chest plate of the nearest <robot> as he spoke, and inserted the two inch sphere that contained the tiny spark of atomic energy that was a robot's life.'

Batteries Power Our Robots

Today, we almost exclusively use batteries to power our robots, but we've improved on those power sources quite a bit since the '40s. Starting back then with heavy lead-acid batteries, we've used Ni-Cad, sealed electrolyte lead-acid, and are working our way to widespread use of nickel metal hydride and lithium-ion polymer batteries. These are just a few of the many secondary or re-chargeable batteries that robot builders use. In primary or non-rechargeable batteries, we first used zinc-carbon 'flashlight' cells and now use alkaline cells, but there are also many other types of primary cells. Many designers feel that robot power is one of the greatest limitations of a really great design, just as it is the foe for electric cars. I personally feel that the greatest advancement needed in mobile robots is battery technology.

Sensors for Asimov's Robots

The following is an excerpt from *Runaround: ".... the two giant robots were invisible but for the dull red of their photoelectric eyes that stared down at them, unblinking, unwavering, and unconcerned."*

Asimov did not go into much detail about his robot's sensory capabilities other than their 'photoelectric eyes.' There are actually many references to photoelectric eyes in his stories, including their dull red appearance. Quite

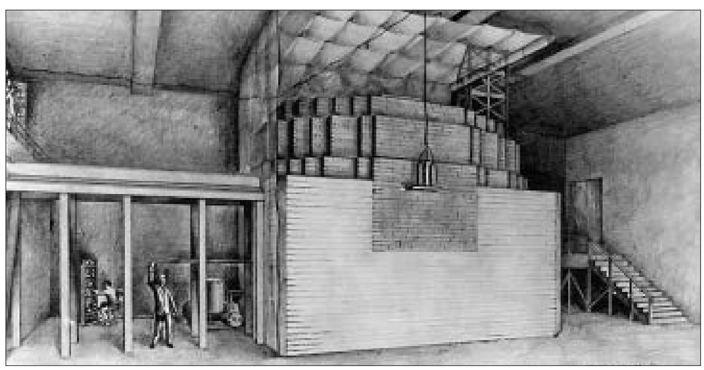


FIGURE 6. Nuclear power in the '40s from the University of Chicago 'atomic pile.'

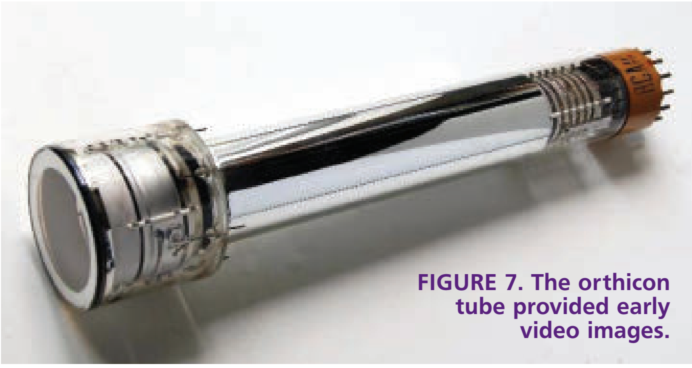


FIGURE 7. The orthicon tube provided early video images.

FIGURE 8. Kawada Industries' Hiro robot.

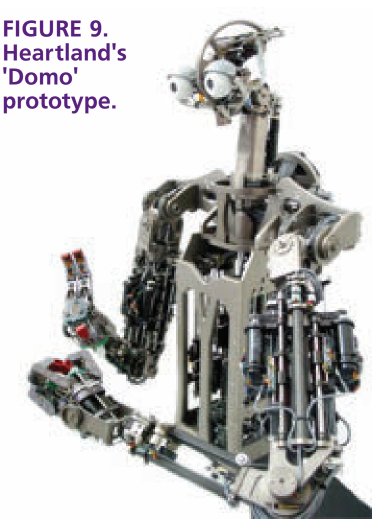
frankly, I'm not sure why a light sensor would put out any color at all. They *receive* light, not *emit* light. (Oh, well.) Red robot eyes have become popular for many robot stories and movie action props. Cylons and Terminators would not have looked half as mean if they did not have red eyes.

Again, the one thing that has stood out to me over the years about Asimov's use of photoelectric cells for robot eyes is that they are not *image* sensors, but are *light* sensors only. Instead of having mega-pixels or even kilo-pixels of resolution, his photoelectric cells had only one pixel. Not much resolution to locate little Gloria, or anything at all. We have to remember that the '40s were the days when television and the iconoscope were just laboratory curiosities. Even the old vidicons or image-orthicons (shown in **Figure 7**) had yet to be invented. As a reader, I just ignored the technical details and let the stories flow through my mind just as they were written.

Robots of Today

Kawada Industries has produced a humanoid industrial robot (shown in **Figure 8**) that is named Hiro. This robot looks nothing like the industrial robots of the past, but more like what I think of as one of Asimov's robots from *Robbie*. Kawada Industries says the robot was designed for human collaboration robotics operating in real life work environments. Hiro's capabilities for this purpose (says the company) include movement, interaction, communication, image recognition, voice recognition, and voice synthesis. The goal: Combine the intelligence of the human being with the characteristics of industrial robots.

FIGURE 9. Heartland's 'Domo' prototype.



Rodney Brooks — ex-professor at MIT and one of the co-founders of iRobot — has also formed a company called Heartland Robotics which is now Rethink Robotics. He and his company are developing a $5,000 robot that is what he calls " ... the robotics industry's version of the iPhone — an affordable enough device that will be intuitive to use, and that will spawn a community of app developers who write software for it. It'll be designed to perform a variety of packaging or light manufacturing tasks. The robot is supposedly capable of being trained to perform a certain repetitive task just by moving its arm and gripper. These will be robots that can perform assembly and packaging tasks that low wage factory workers do today." The robot depicted in **Figure 9** has been suggested to be similar to what Brooks is proposing, but certainly not nearly as complex and ungainly.

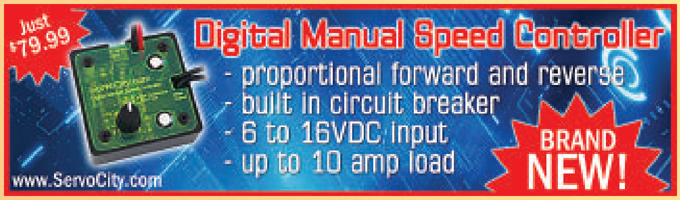
Final Thoughts

Kawada, Rethink Robotics, Redwood Robotics, and other companies are hard at work on humanoid robot designs that will be able to work alongside workers in all types of industries. Today's Microsoft Kinect and Asus Xtion, and other ingenious devices give our machines vision and the ability to actually interpret what they see. LiPo batteries power our robots, and their brains consist of microcontrollers and microprocessors that were not even dreamed of 50 years ago. Bipedal humanoids, space robots, and military machines have long surpassed industrial robots with their incredible capabilities.

Isaac Asimov would be quite proud of the inspiring influence that he has had on the science of robotics. Maybe Honda's Asimo shown in **Figure 10** represents this influence best. Though the acronym *supposedly* came from *Advanced Step in Innovative Mobility*, all of us really know just how influential this great writer has been. **SV**

FIGURE 10. Honda's Asimo.

ROBO-LINKS



For the finest in robots, parts, and services, go to www.servomagazine.com and click on Robo-Links.

Advertiser Index

Prices and availability subject to change without notice.